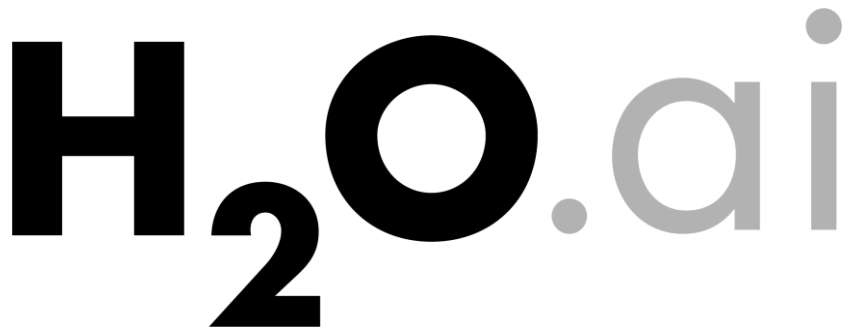
Project "Deep Water"

H_2O's Integration with TensorFlow



Jo-fai (Joe) Chow

Data Scientist

joe@h2o.ai

@matlabulous

TensorFlow Paris Meetup

30th November, 2016

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 H2O in 2014
- Data Scientist
  - From 2015
  - Virgin Media (UK)
  - Domino Data Lab (Silicon Valley)
  - H_2O.ai (Silicon Valley)

H_2O.ai

Agenda

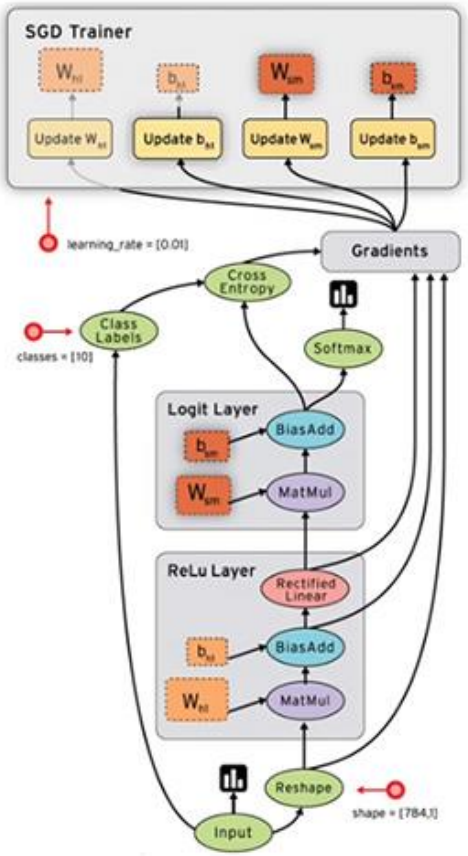
- Introduction
  - About TensorFlow
  - TensorFlow Use Cases
  - About H_2O.ai
- Project Deep Water
  - Motivation
  - Benefits
  - H_2O + TensorFlow Live Demo
- Conclusions

H_2O.ai

About TensorFlow

About TensorFlow

- Open source machine learning framework by Google
- Python / C++ API
- TensorBoard
  - Data Flow Graph Visualization
- Multi CPU / GPU
  - v0.8+ distributed machines support
- Multi devices support
  - desktop, server and Android devices
- Image, audio and NLP applications
- HUGE Community
- Support for Spark, Windows …



TensorFlow Wrappers

- TFLearn – Simplified interface
- keras – TensorFlow + Theano
- tensorflow.rb – Ruby wrapper
- TensorFlow.jl – Julia wrapper
- TensorFlow for R – R wrapper

- … and many more!
- See: github.com/jtoy/awesome-tensorflow

H2O.ai

~~TensorFlow Use Cases~~

Very Cool TensorFlow Use Cases

Some of the cool things you can do with TensorFlow

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.

Iterations
000,084

Learning rate: 0.03

Activation: Tanh

Regularization: None

Regularization rate: 0

Problem type: Classification

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

X_1X_2

$\sin(X_1)$

$\sin(X_2)$

\+ − 2 HIDDEN LAYERS

4 neurons

2 neurons

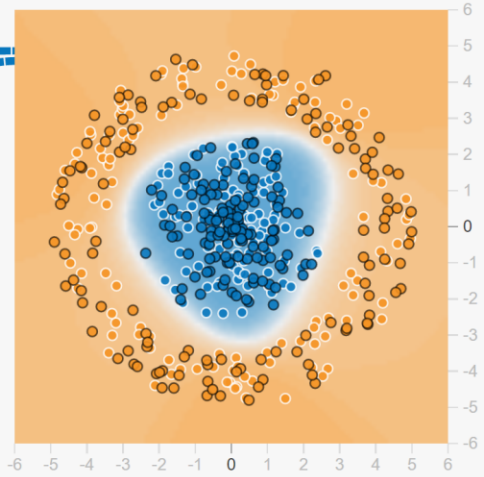
*This is the output from one **neuron**. Hover to see it larger.*

*The outputs are mixed with varying **weights**, shown by the thickness of the lines.*

OUTPUT

Test loss 0.020

Training loss 0.023



Colors shows data, neuron and weight values.

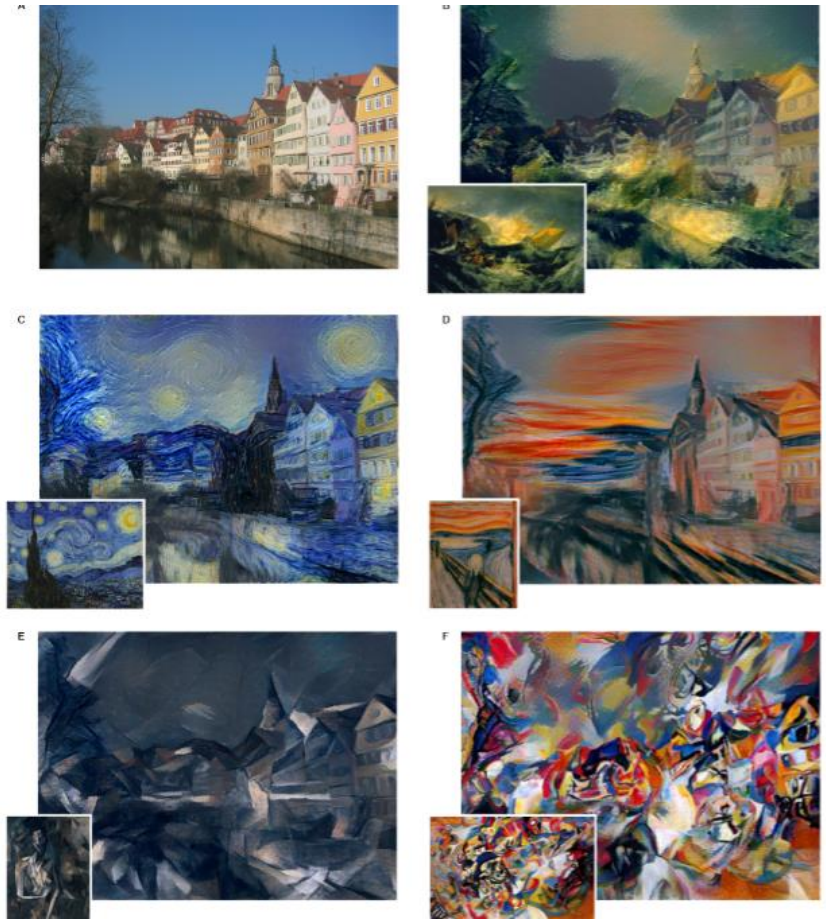
-1 0 1

☑ Show test data ☐ Discretize output

playground.tensorflow.org

Neural Style Transfer in TensorFlow

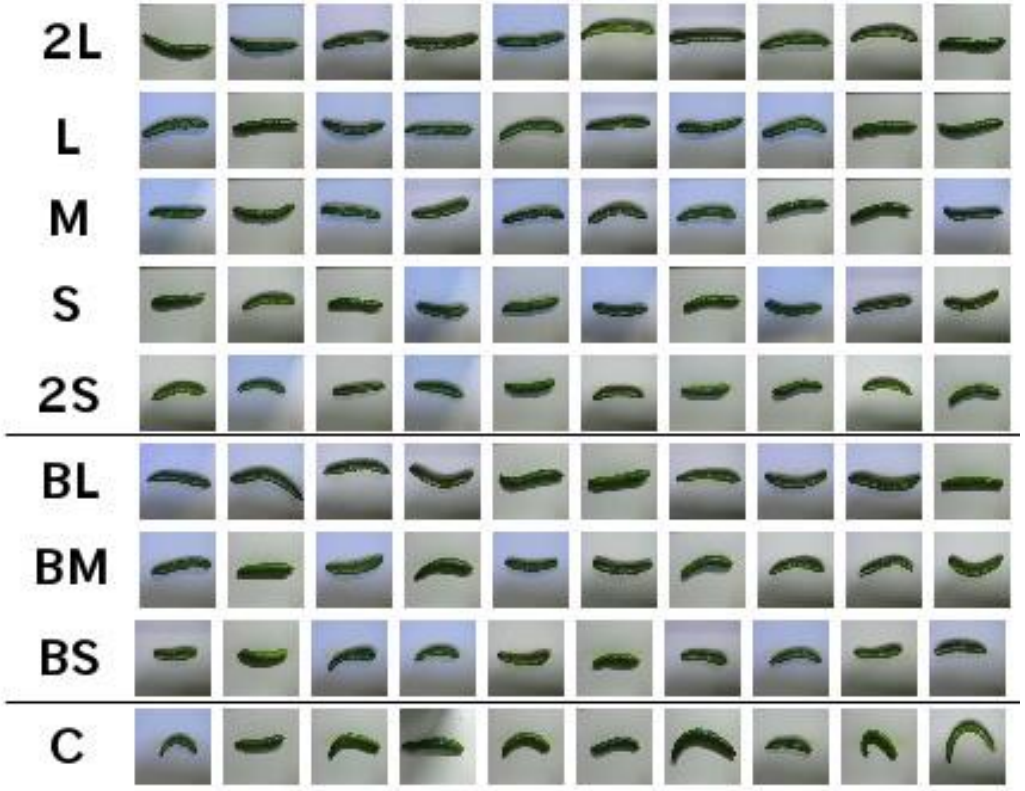
- Neural Style
  - "… a technique to train a deep neural network to separate artistic style from image structure, and combine the style of one image with the structure of another"
- Original Paper
  - "A Neural Algorithm of Artistic Style"
- TensorFlow Implementation
  - [Link]



Sorting Cucumbers

- Problem
  - Sorting cucumbers is a laborious process.
  - In a Japanese farm, the farmer's wife can spend up to eight hours a day sorting cucumbers during peak harvesting period.
- Solution
  - Farmer's son (Makoto Koike) used TensorFlow, Arduino and Raspberry Pi to create an automatic cucumber sorting system.

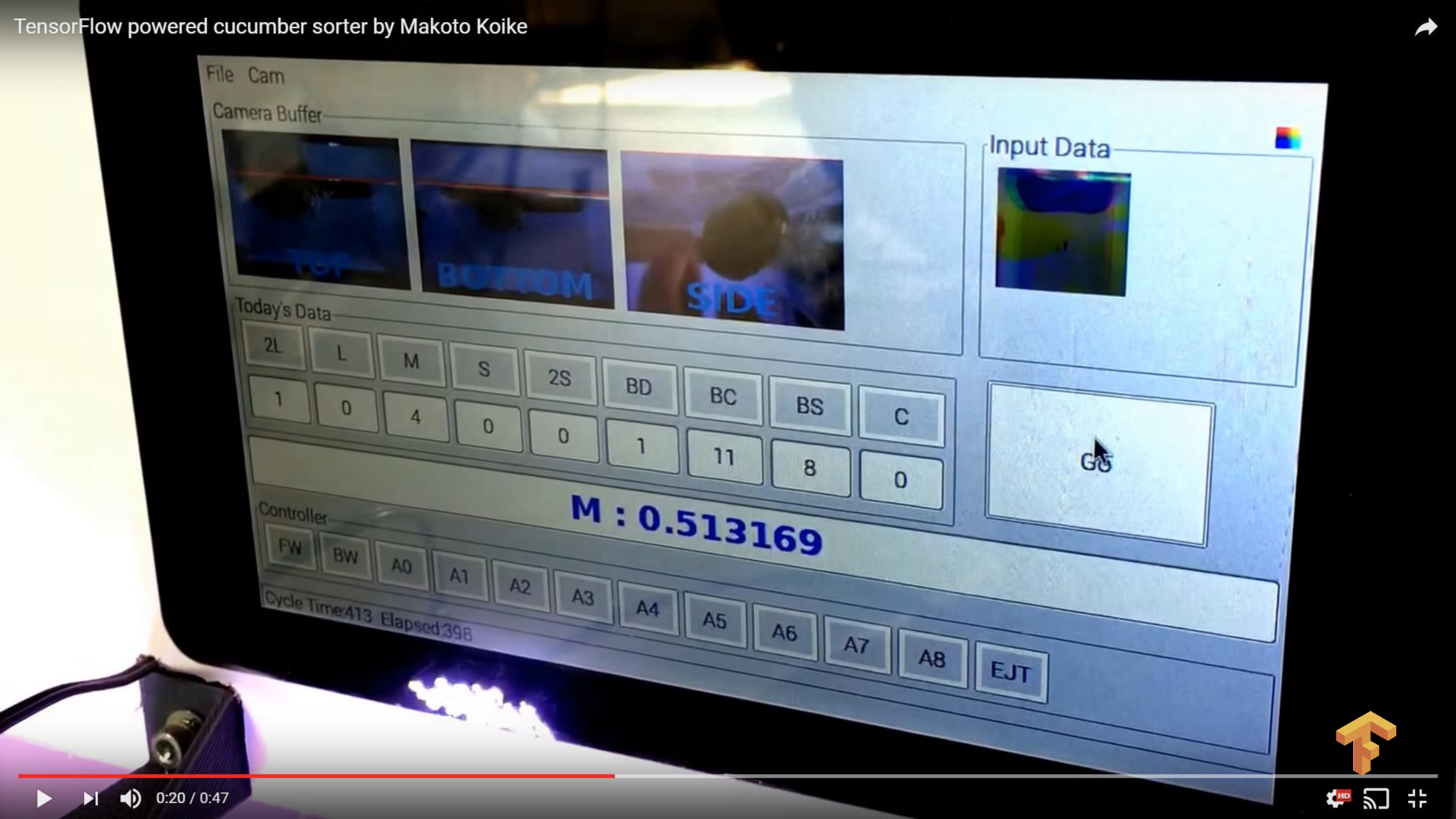
Sorting Cucumbers



- Classification Problem
  - Input: cucumber photos (side, top, bottom)
  - Output: one of nine classes

- Google's Blog Post [Link]
- YouTube Video [Link]


TensorFlow powered cucumber sorter by Makoto Koike
0:02 / 0:47


TensorFlow powered cucumber sorter by Makoto Koike
File Cam
Camera Buffer
TOP
BOTTOM
SIDE
Input Data
Today's Data
2L L M S 2S BD BC BS C
1 0 4 0 0 1 11 8 0
GO
M : 0.513169
Controller
FW BW A0 A1 A2 A3 A4 A5 A6 A7 A8 EJT
Cycle Time:413 Elapsed:398
0:20 / 0:47


TensorFlow powered cucumber sorter by Makoto Koike
宅急便
Bou
0:10 / 0:47

Of course
there are more TensorFlow use cases

The key message here is …

TensorFlow

democratizes

the power of deep learning

About H_2O.ai

What exactly is H_2O?

Company Overview

Founded	2011 Venture-backed, debuted in 2012
Products	• H2O 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




H2O.ai

H_2O is an open source platform empowering business transformation

Bring AI To Business Empower Transformation

Financial Services, Insurance and Healthcare as Our Vertical Focus

Community as Our Foundation

Users In Various Verticals Adore H_2O

H2O.ai

H2O In Action

www.h2o.ai/customers

Capital One

Capital One uses H2O open source machine learning for various use cases.

MarketShare

H2O predictive analytics helps boost the impact and results of digital marketing.

Kaiser

Kaiser uses H2O machine learning to save lives.

Zurich Insurance

Zurich turned to H2O as a strategic differentiator for commercial insurance.

Progressive

Progressive uses H2O predictive analytics for user-based insurance.

Comcast

Comcast uses H2O to improve customer experience.

Hospital Corporation of America

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

McKesson

McKesson discusses the adoption of artificial intelligence in healthcare.

Macy's

Macy's uses H2O for personalized site recommendations.

Transamerica

Transamerica turns to H2O to develop a product recommendation platform for insurance.

Paypal

Paypal turned to H2O Deep Learning for fraud detection and customer churn.

eBay

eBay chose H2O for open source machine learning.

H2O.ai

H_2O.ai Makes A Difference as an AI Platform

Open Source

- 100% open source

Big Data Ecosystem

hadoop

Spark

Flexible Interface

R

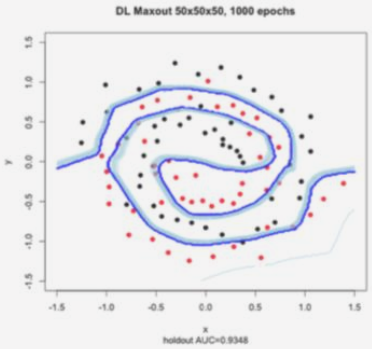
Scala

Java

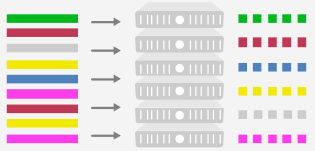
python

H_2O Flow

Smart and Fast Algorithms



Scalability and Performance



- Distributed In-Memory Computing Platform
- Distributed Algorithms
- Fine-Grain MapReduce

Rapid Model Deployment

- Highly portable models deployed in Java (POJO) and Model Object Optimized (MOJO)
- Automated and streamlined scoring service deployment with Rest API

REST API

GPU Enablement

GPU

Cloud Integration

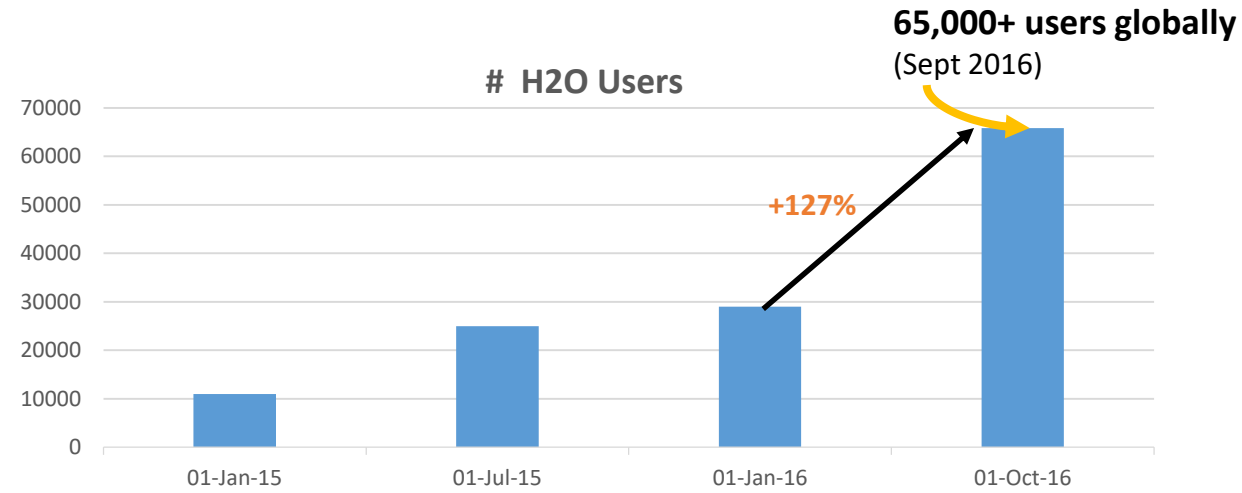
amazon web services™

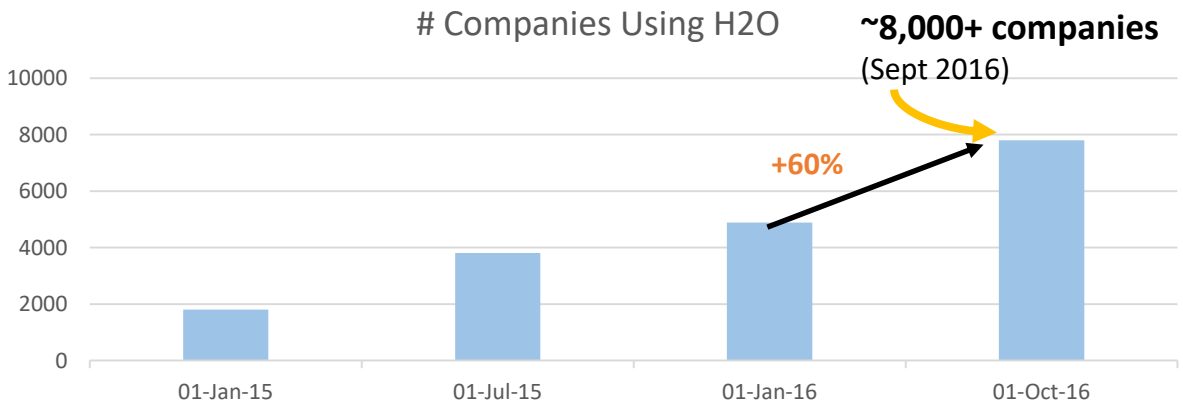
Microsoft Azure

Google

H_2O Community Growth

Tremendous Momentum Globally





* DATA FROM GOOGLE ANALYTICS EMBEDDED IN THE END USER PRODUCT

Large User Circle

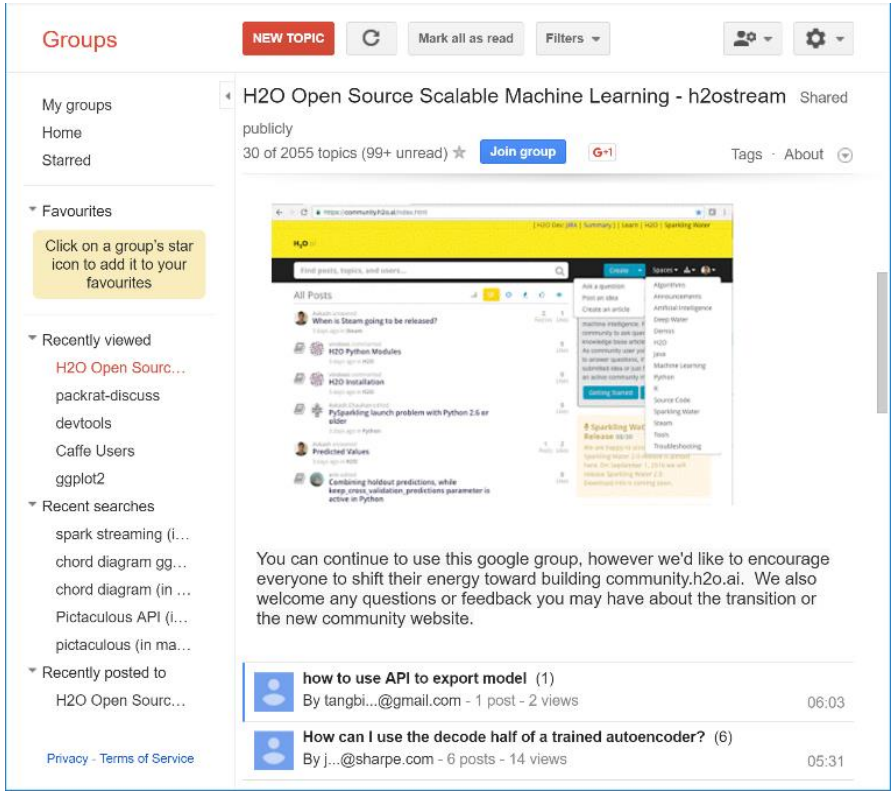
- 65,000+ users from ~8,000 companies in 140 countries. Top 5 from:

1. United States
2. India
3. Japan
4. Germany
5. United Kingdom

H2O.ai

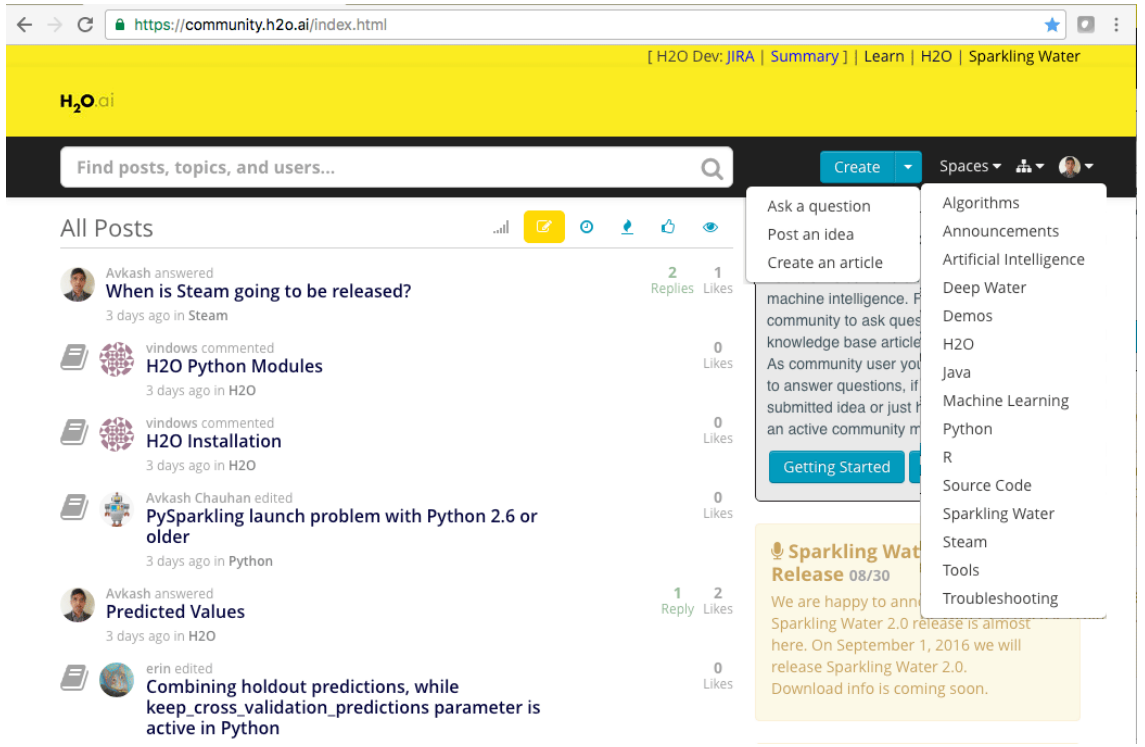
H_2O Community Support

Google forum – h2osteam



Please try

community.h2o.ai



H_2O for Kaggle Competitions

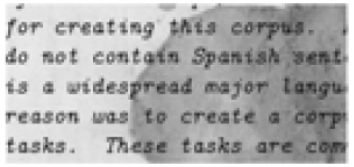
CIFAR-10 Competition Winners: Interviews with Dr. Ben Graham, Phil Culliton, & Zygmunt Zając

Triskelion | 01.02.2015

READ MORE

"I did really like H2O's deep learning implementation in R, though - the interface was great, the back end extremely easy to understand, and it was scalable and flexible. Definitely a tool I'll be going back to."

Kaggle challenge 2nd place winner Colin Priest



Completed • Knowledge • 161 teams

Denoising Dirty Documents

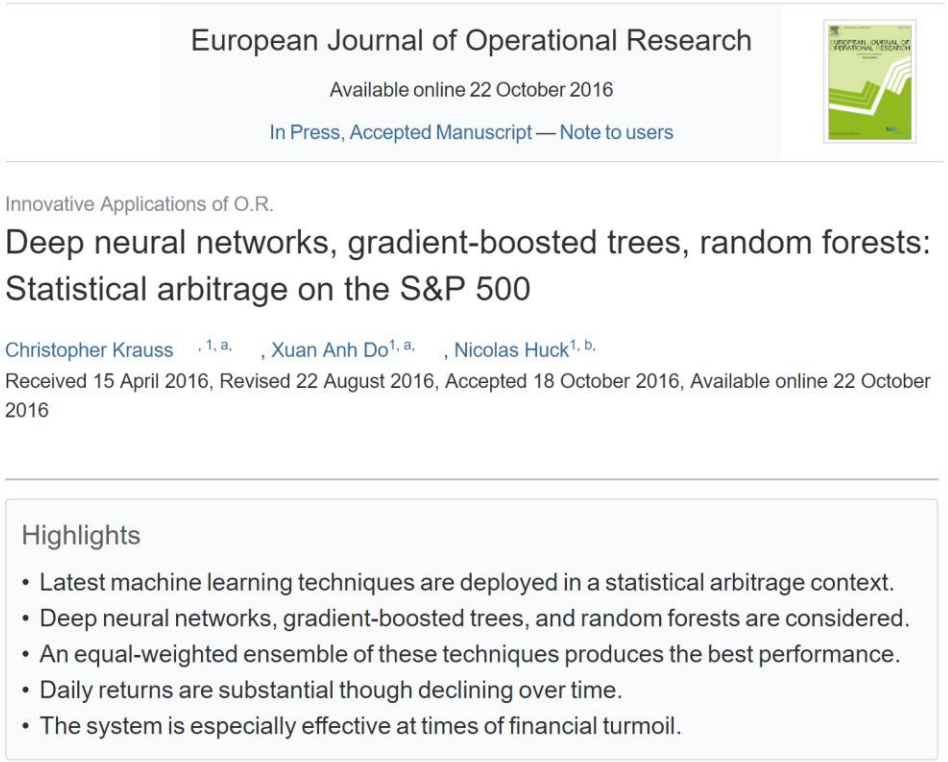
Mon 1 Jun 2015 – Mon 5 Oct 2015 (3 months ago)

READ MORE

"For my final competition submission I used an ensemble of models, including 3 deep learning models built with R and h2o."

H_2O.ai

H_2O for Academic Research



European Journal of Operational Research

Available online 22 October 2016

In Press, Accepted Manuscript — Note to users

Innovative Applications of O.R.

Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500

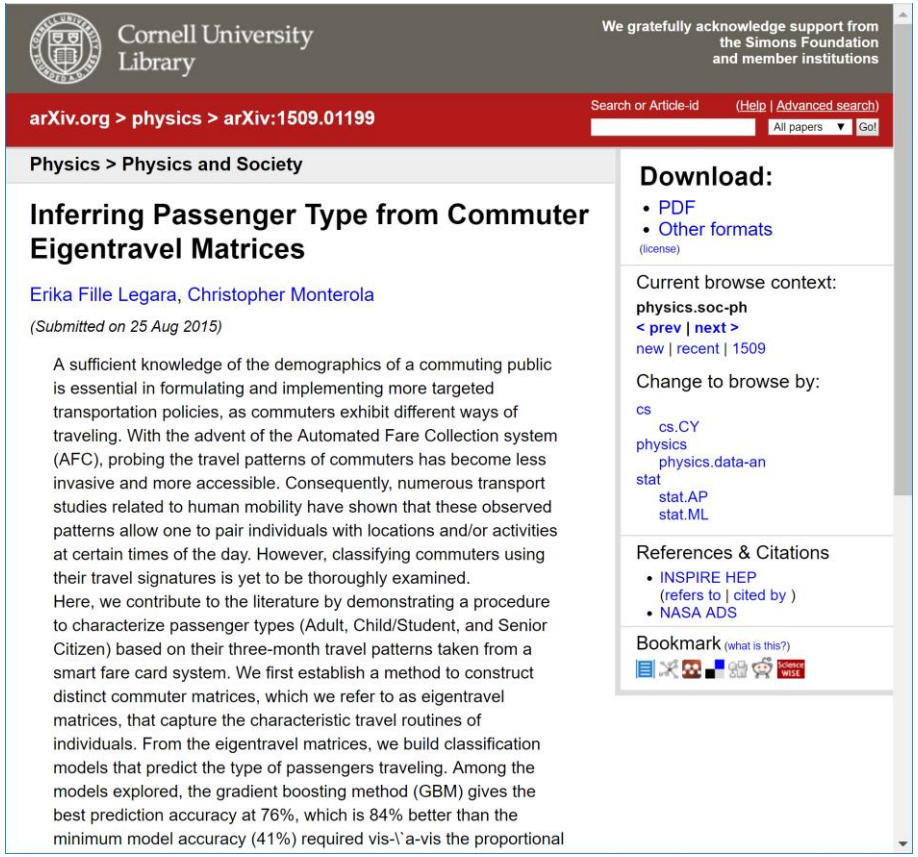
Christopher Krauss [1, a], Xuan Anh Do [1, a], Nicolas Huck [1, b]

Received 15 April 2016, Revised 22 August 2016, Accepted 18 October 2016, Available online 22 October 2016

Highlights

- Latest machine learning techniques are deployed in a statistical arbitrage context.
- Deep neural networks, gradient-boosted trees, and random forests are considered.
- An equal-weighted ensemble of these techniques produces the best performance.
- Daily returns are substantial though declining over time.
- The system is especially effective at times of financial turmoil.

http://www.sciencedirect.com/science/article/pii/S0377221716308657



Cornell University Library

We gratefully acknowledge support from the Simons Foundation and member institutions

arXiv.org > physics > arXiv:1509.01199

Search or Article-id (Help | Advanced search) All papers Go!

Physics > Physics and Society

Inferring Passenger Type from Commuter Eigentravel Matrices

Erika Fille Legara, Christopher Monterola

(Submitted on 25 Aug 2015)

A sufficient knowledge of the demographics of a commuting public is essential in formulating and implementing more targeted transportation policies, as commuters exhibit different ways of traveling. With the advent of the Automated Fare Collection system (AFC), probing the travel patterns of commuters has become less invasive and more accessible. Consequently, numerous transport studies related to human mobility have shown that these observed patterns allow one to pair individuals with locations and/or activities at certain times of the day. However, classifying commuters using their travel signatures is yet to be thoroughly examined. Here, we contribute to the literature by demonstrating a procedure to characterize passenger types (Adult, Child/Student, and Senior Citizen) based on their three-month travel patterns taken from a smart fare card system. We first establish a method to construct distinct commuter matrices, which we refer to as eigentravel matrices, that capture the characteristic travel routines of individuals. From the eigentravel matrices, we build classification models that predict the type of passengers traveling. Among the models explored, the gradient boosting method (GBM) gives the best prediction accuracy at 76%, which is 84% better than the minimum model accuracy (41%) required vis-\`a-vis the proportional

Download:

- PDF
- Other formats

(license)

Current browse context: physics.soc-ph

< prev | next >

new | recent | 1509

Change to browse by:

cs
cs.CY
physics
physics.data-an
stat
stat.AP
stat.ML

References & Citations

- INSPIRE HEP (refers to | cited by)
- NASA ADS

Bookmark (what is this?)

https://arxiv.org/abs/1509.01199

H_2O.ai

H_2O

democratizes

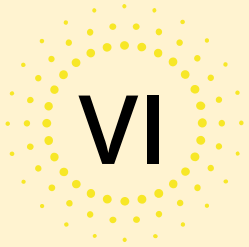
artificial intelligence & big data science

Our Open Source Products

100% Open Source. Big Data Science for Everyone!

H_2O.ai Offers AI Open Source Platform

Product Suite to Operationalize Data Science with Visual Intelligence



Visual Intelligence and UX Framework For Data Interpretation and Story Telling on top of Beautiful Data Products

100% Open Source

H_2O.ai

In-Memory, Distributed Machine Learning Algorithms with Speed and Accuracy

Deep Water

State-of-the-art Deep Learning on GPUs with TensorFlow, MXNet or Caffe with the ease of use of H2O

Spark + H_2O

SPARKLING WATER

H2O Integration with Spark. Best Machine Learning on Spark.

Steam

Operationalize and Streamline Model Building, Training and Deployment Automatically and Elastically

H_2O.ai Offers AI Open Source Platform

Product Suite to Operationalize Data Science with Visual Intelligence

Visual Intelligence and UX Framework For Data Interpretation and Story Telling on top of Beautiful Data Products

100% Open Source

H_2O.ai

In-Memory, Distributed Machine Learning Algorithms with Speed and Accuracy

Deep Water

State-of-the-art Deep Learning on GPUs with TensorFlow, MXNet or Caffe with the ease of use of H2O

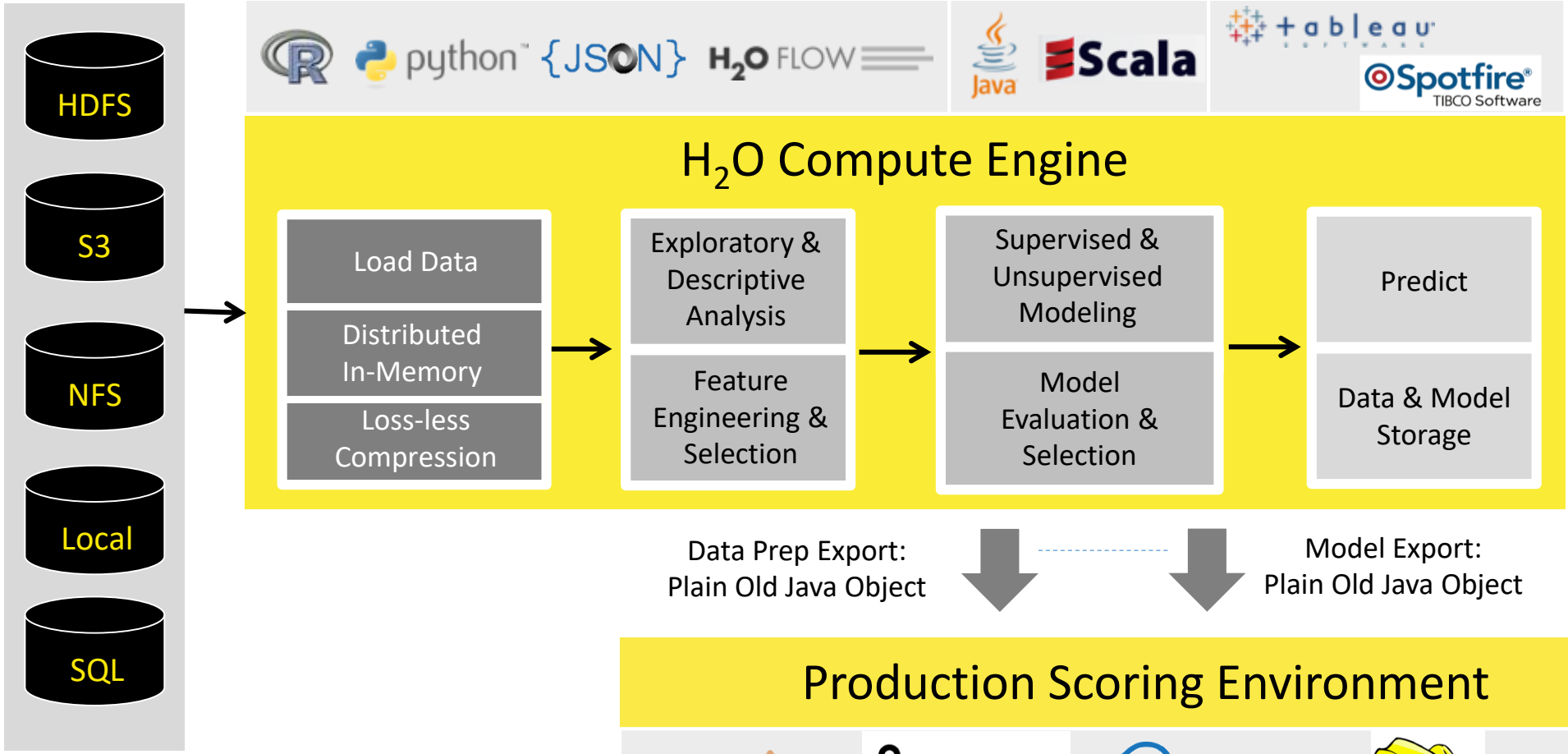
Spark + H_2O SPARKLING WATER

H2O Integration with Spark. Best Machine Learning on Spark.

Steam

Operationalize and Streamline Model Building, Training and Deployment Automatically and Elastically

High Level Architecture



Algorithms Overview

Supervised Learning

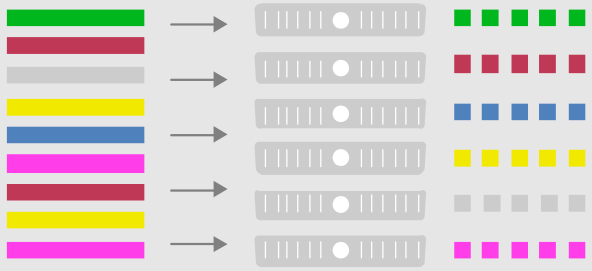
Statistical Analysis	• **Generalized Linear Models**: Binomial, Gaussian, Gamma, Poisson and Tweedie • **Naïve Bayes**
Ensembles	• **Distributed Random Forest**: Classification or regression models • **Gradient Boosting Machine**: Produces an ensemble of decision trees with increasing refined approximations
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

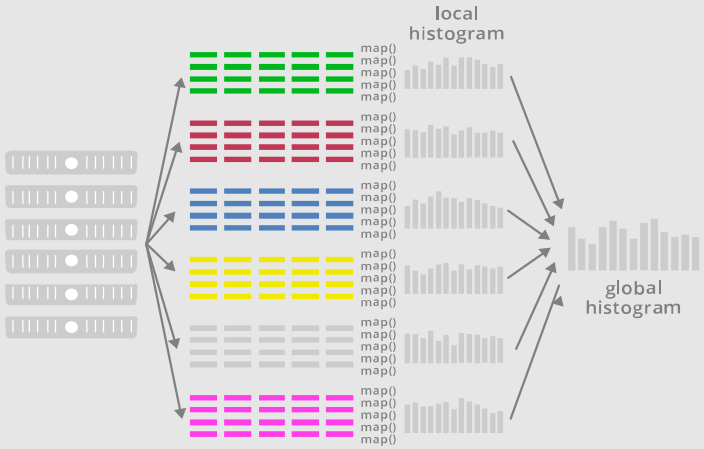
Clustering	• **K-means**: Partitions observations into k clusters/groups of the same spatial size. Automatically detect optimal k
Dimensionality Reduction	• **Principal Component Analysis**: Linearly transforms correlated variables to independent components • **Generalized Low Rank Models:** extend the 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 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 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%

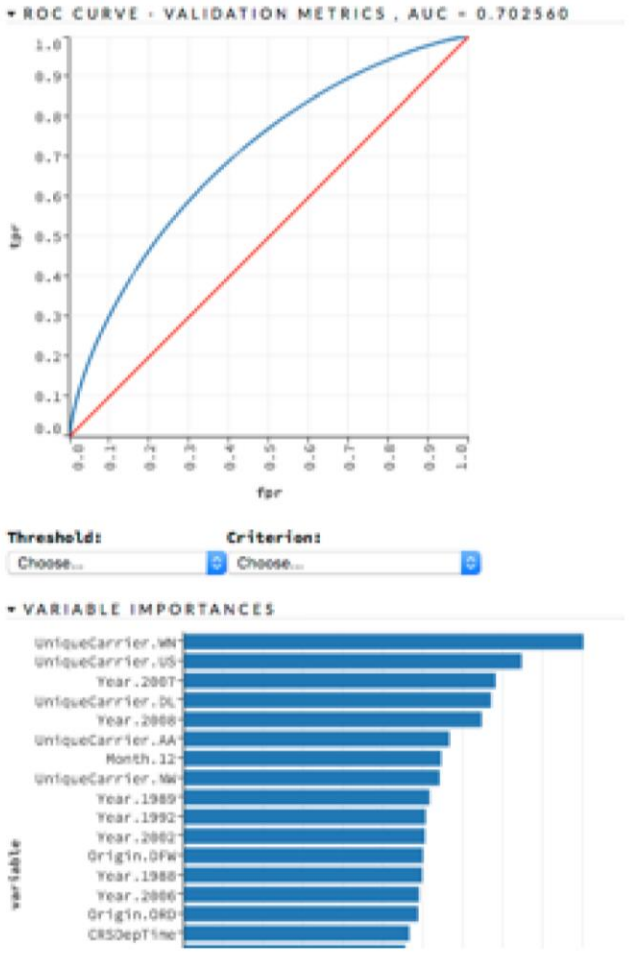
Iterations: 12. Epochs: 0.628821. Speed: 2,243,735 samples/sec. Estimated time left: 21.849 sec

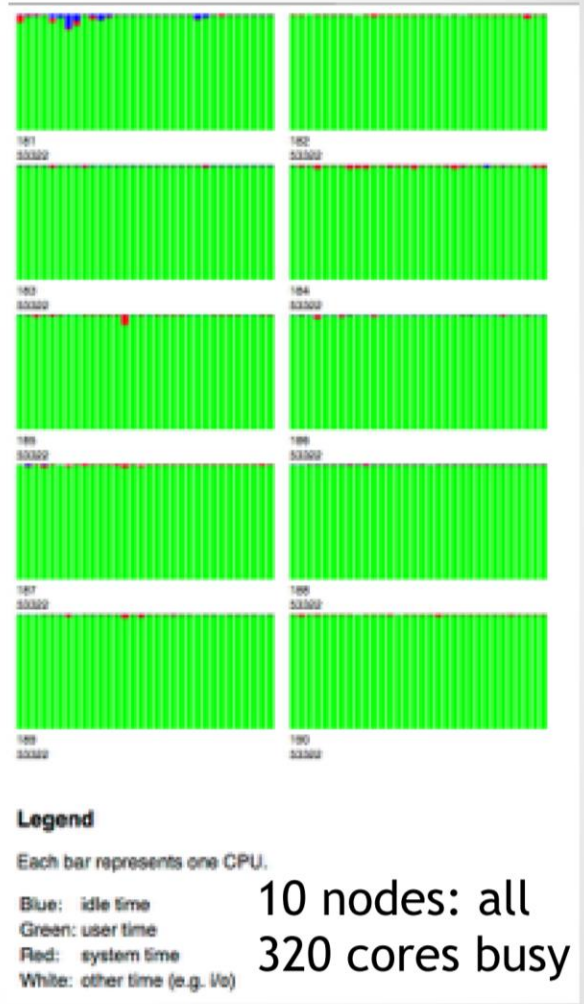
Actions: View | Cancel Job

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**

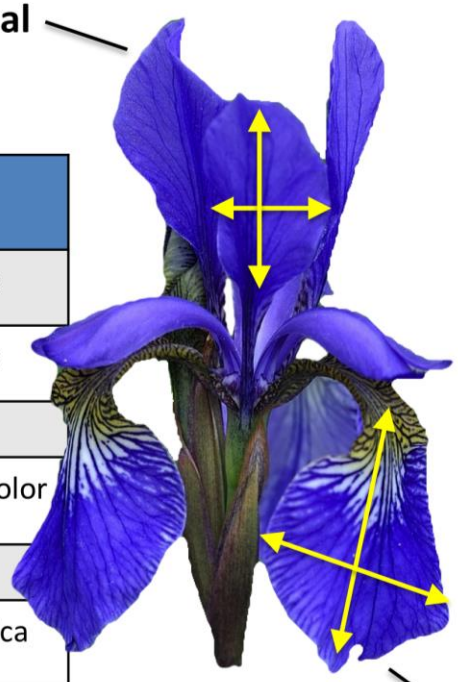
H_2O in Action

Quick Demo (5 minutes)

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)

H_2O + R

h2o_iris_demo.R

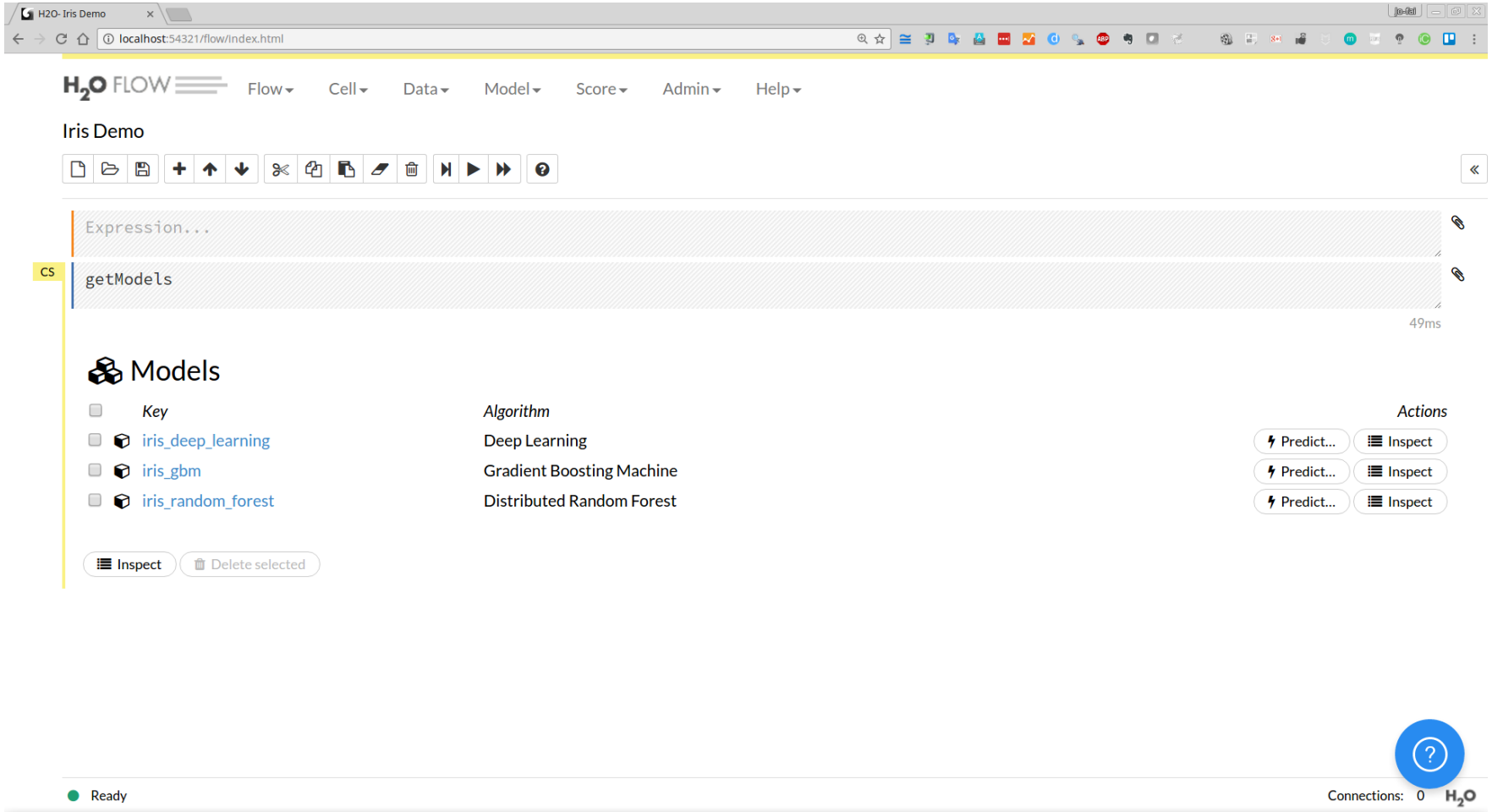
Source on Save | Run | Source

```r
# ------------------------------------------------------------------------------
# Build a simple classification model using iris dataset
# ------------------------------------------------------------------------------

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

# Import data from a R data frame
data(iris)
d_iris <- as.h2o(iris)

# Define Targets and Features
target <- "Species"
features <- setdiff(colnames(d_iris), c("Species"))

# ------------------------------------------------------------------------------
# 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)
```

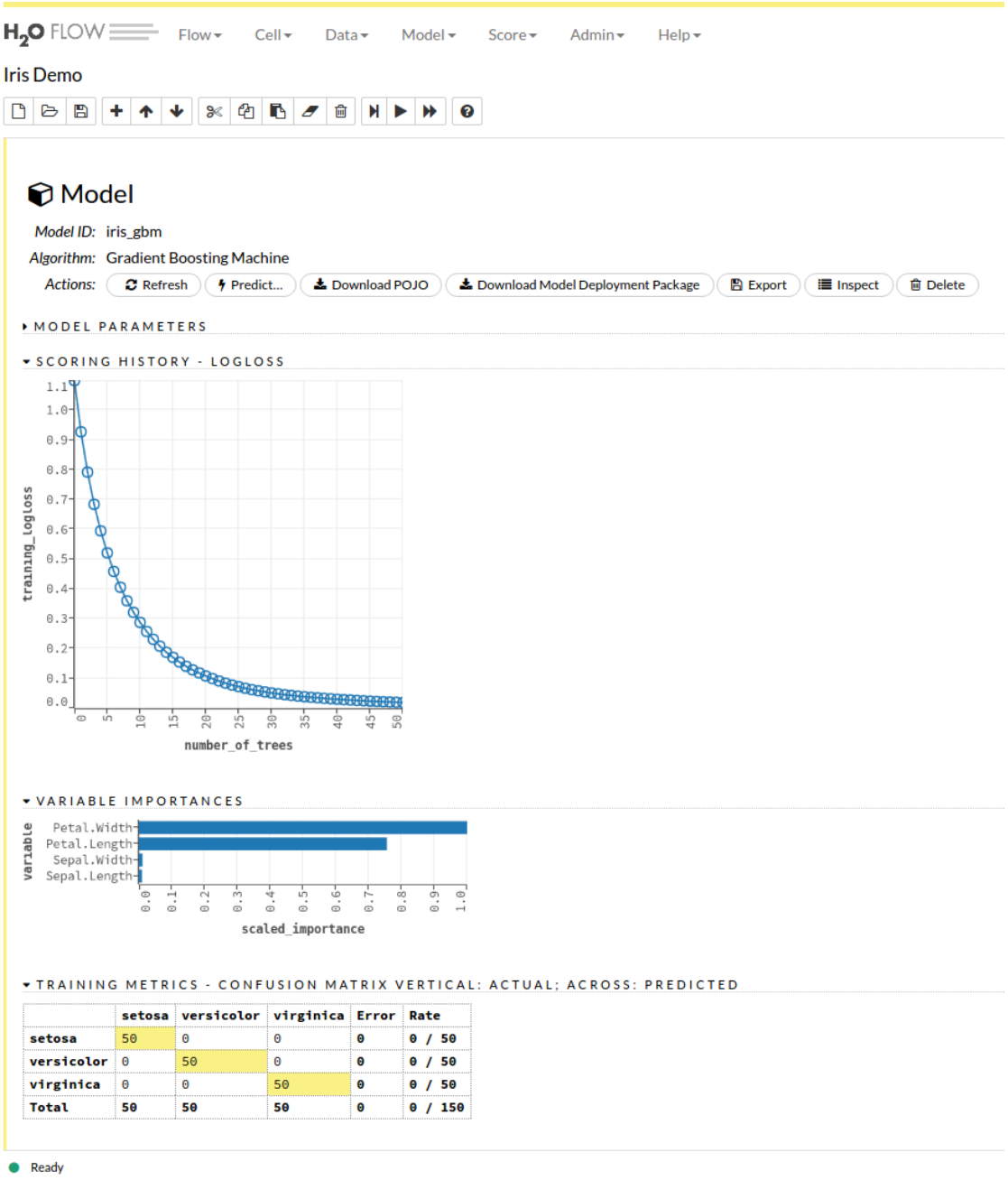
Please try

H_2O.ai

H_2O Flow (Web)


H2O- Iris Demo
localhost:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Iris Demo
Expression...
CS
getModels
49ms
Models
Key
Algorithm
Actions
iris_deep_learning
Deep Learning
Predict...
Inspect
iris_gbm
Gradient Boosting Machine
Predict...
Inspect
iris_random_forest
Distributed Random Forest
Predict...
Inspect
Inspect
Delete selected
Ready
Connections: 0
H2O

H_2O Flow

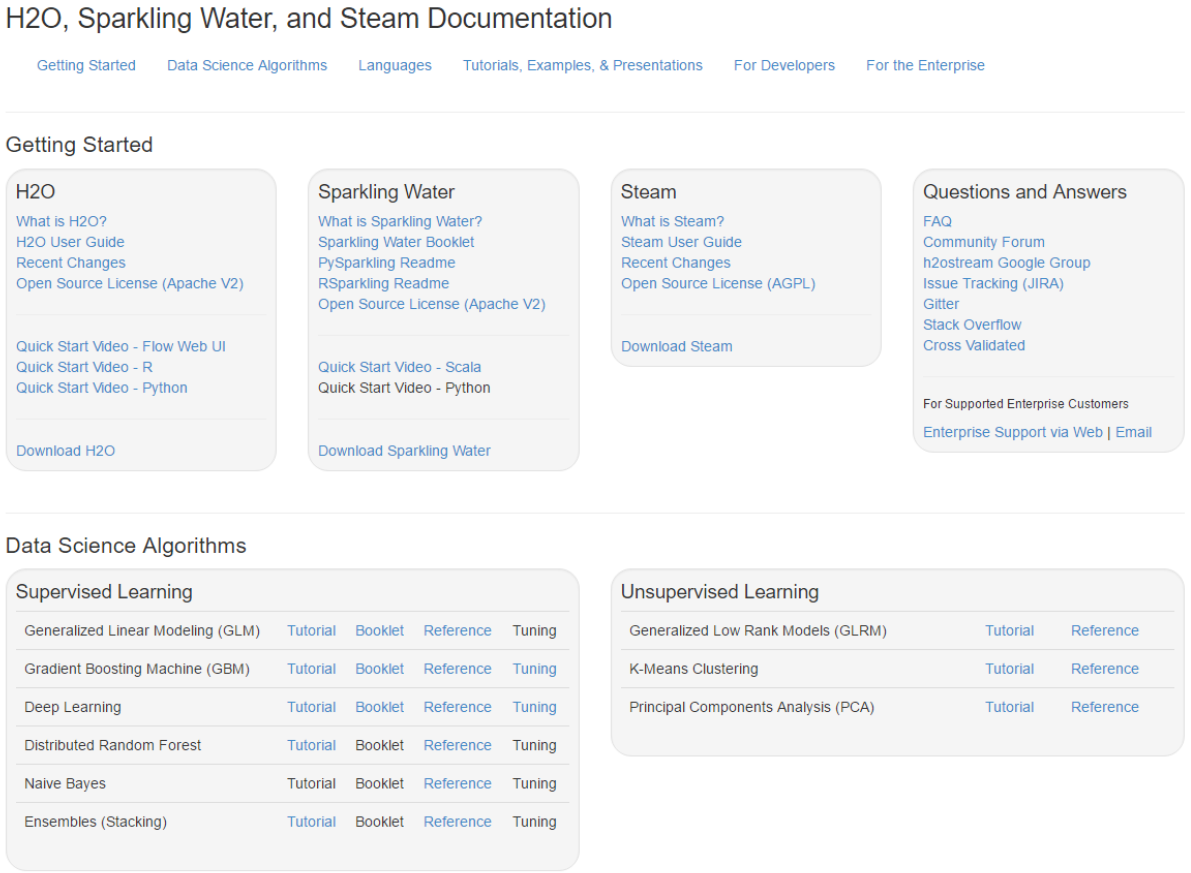


	setosa	versicolor	virginica	Error	Rate
setosa	50	0	0	0	0 / 50
versicolor	0	50	0	0	0 / 50
virginica	0	0	50	0	0 / 50
Total	50	50	50	0	0 / 150

H2O.ai

Key Learning Resources

- Help Documentations
  - docs.h2o.ai
- Meetups
  - bit.ly/h2o_meetups
- YouTube Channel
  - bit.ly/h2o_youtube



H2O, Sparkling Water, and Steam Documentation

Getting Started | Data Science Algorithms | Languages | Tutorials, Examples, & Presentations | For Developers | For the Enterprise

Getting Started

H2O
What is H2O?
H2O User Guide
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
RSparkling Readme
Open Source License (Apache V2)

Quick Start Video - Scala
Quick Start Video - Python

Download Sparkling Water

Steam
What is Steam?
Steam User Guide
Recent Changes
Open Source License (AGPL)

Download Steam

Questions and Answers
FAQ
Community Forum
h2ostream Google Group
Issue Tracking (JIRA)
Gitter
Stack Overflow
Cross Validated

For Supported Enterprise Customers
Enterprise Support via Web | Email

Data Science 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
Ensembles (Stacking)	Tutorial	Booklet	Reference	Tuning

Unsupervised Learning

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

H_2O.ai Offers AI Open Source Platform

Product Suite to Operationalize Data Science with Visual Intelligence

Visual Intelligence and UX Framework For Data Interpretation and Story Telling on top of Beautiful Data Products

100% Open Source

H_2O.ai

In-Memory, Distributed Machine Learning Algorithms with Speed and Accuracy

Deep Water

State-of-the-art Deep Learning on GPUs with TensorFlow, MXNet or Caffe with the ease of use of H2O

Spark + H_2O SPARKLING WATER

H2O Integration with Spark. Best Machine Learning on Spark.

Steam

Operationalize and Streamline Model Building, Training and Deployment Automatically and Elastically

Both TensorFlow and H_2O are widely used

The usage of Hadoop/Big Data tools grew to 39%, up from 29% in 2015 (and 17% in 2014), driven by Apache Spark, MLlib (Spark Machine Learning Library) and H2O.
See also

- KDnuggets interview with Spark Creator Matei Zaharia
- KDnuggets interview with Arno Candel, H2O.ai on How to Quick Start Deep Learning with H2O

http://www.kdnuggets.com

H2O and TensorFlow are tied

usage of Deep Learning tools in past year

Deep Learning Tools & Platforms: Caffe (63)	0%
Cuda-convnet (22)	0%
Convnet.js (8)	0%
carch (2)	0%
Deeplearning4j (46)	0%
H2O (190)	1%
MATLAB Deep Learning Toolbox (57)	0%
Microsoft CNTK (25)	0%
mxnet (16)	0%
Nervana (2)	0%
Tensorflow (190)	1%
Theano ecosystem including Keras, Lasagne, Pylearn2 (140)	1%
Torch (28)	0%
Veles (2)	0%
Other Deep Learning Tools (104)	1%

H_2O.ai

TensorFlow democratizes the power of deep learning.

H_2O democratizes artificial intelligence & big data science.

There are other open source libraries like MXNet and Caffe too.

Let's have a party, this will be fun!

H_2O.ai

Deep Water

Next-Gen Distributed Deep Learning with H_2O

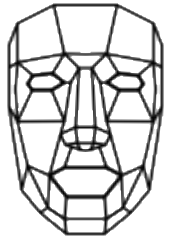
One Interface - GPU Enabled - Significant Performance Gains

Inherits All H_2O Properties in Scalability, Ease of Use and Deployment

TensorFlow Caffe H_2O.ai mxnet

Deep Water

H_2O integrates with existing GPU backends for significant performance gains



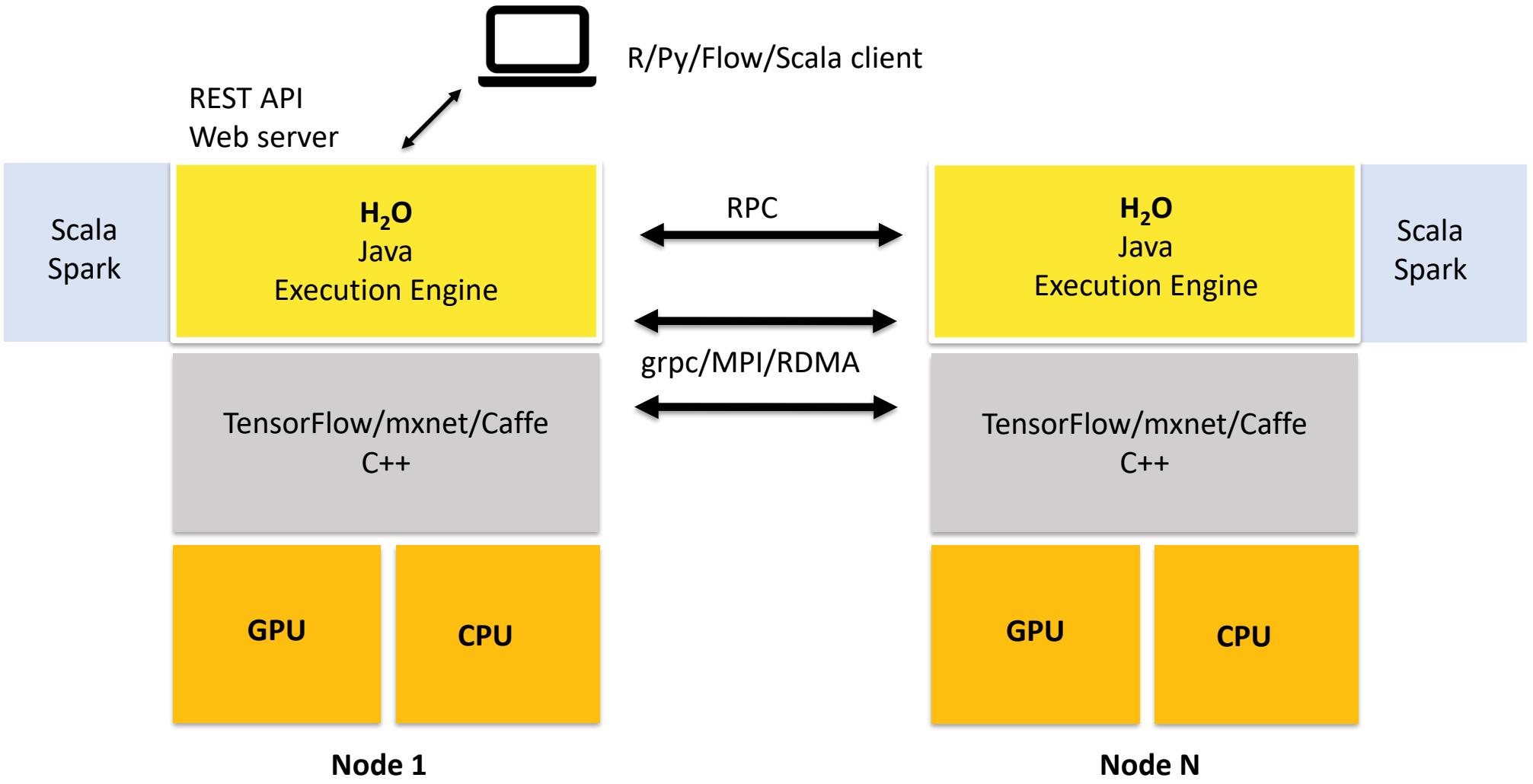
Convolutional Neural Networks enabling **Image, video, speech recognition**

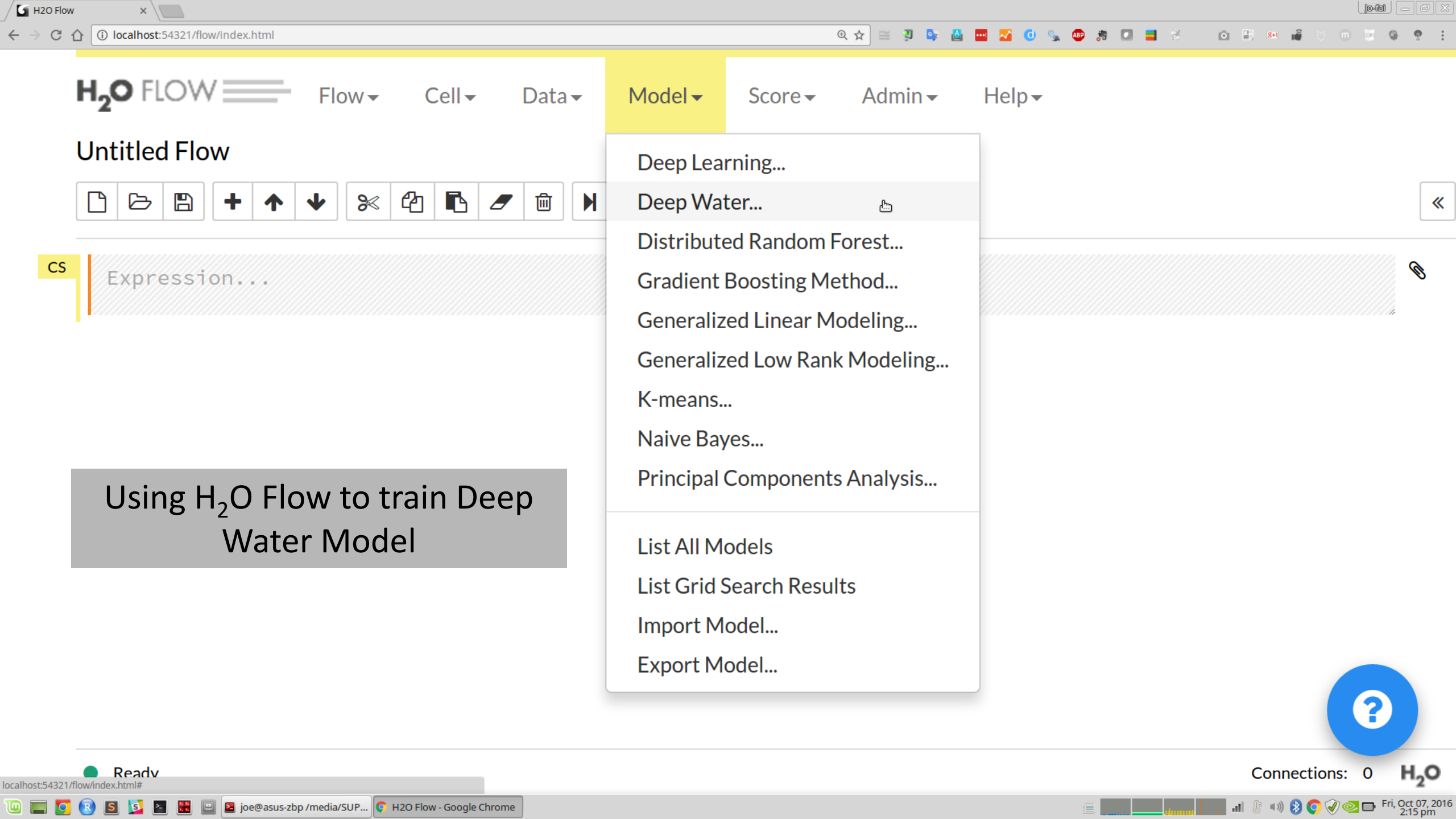
Recurrent Neural Networks enabling **natural language processing, sequences, time series,** and more

Hybrid Neural Network Architectures enabling **speech to text translation, image captioning, scene parsing** and more

H_2O.ai

Deep Water Architecture




H2O Flow
localhost:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Untitled Flow
CS
Expression...
Deep Learning...
Deep Water...
Distributed Random Forest...
Gradient Boosting Method...
Generalized Linear Modeling...
Generalized Low Rank Modeling...
K-means...
Naive Bayes...
Principal Components Analysis...
List All Models
List Grid Search Results
Import Model...
Export Model...
Using H2O Flow to train Deep Water Model
Ready
Connections: 0
H2O
localhost:54321/flow/index.html#
joe@asus-zbp /media/SUP...
H2O Flow - Google Chrome
Fri, Oct 07, 2016
2:15 pm

Same H2O R/Python Interface

To build a LeNet image classification model in H2O, simply specify network = "lenet":

```
model <- h2o.deepwater(x=path, y=response,
                       training_frame=df, epochs=50,
                       learning_rate=1e-3, network = "lenet")
model
```

```
  |======================================================================| 100%
Model Details:
==============

H2OMultinomialModel: deepwater
Model ID:  DeepWater_model_R_1477378862430_2
Status of Deep Learning Model: lenet, 1.6 MB, predicting C2, 3-class classification, 14,336 training sampl
es, mini-batch size 32
  input_neurons     rate momentum
1          2352 0.000986 0.990000


H2OMultinomialMetrics: deepwater
** Reported on training data. **
** Metrics reported on full training frame **

Training Set Metrics:
=====================

Extract training frame with `h2o.getFrame("cat_dog_mouse.hex_sid_95f8_1")`
MSE: (Extract with `h2o.mse`) 0.131072
RMSE: (Extract with `h2o.rmse`) 0.3620386
Logloss: (Extract with `h2o.logloss`) 0.4176429
```

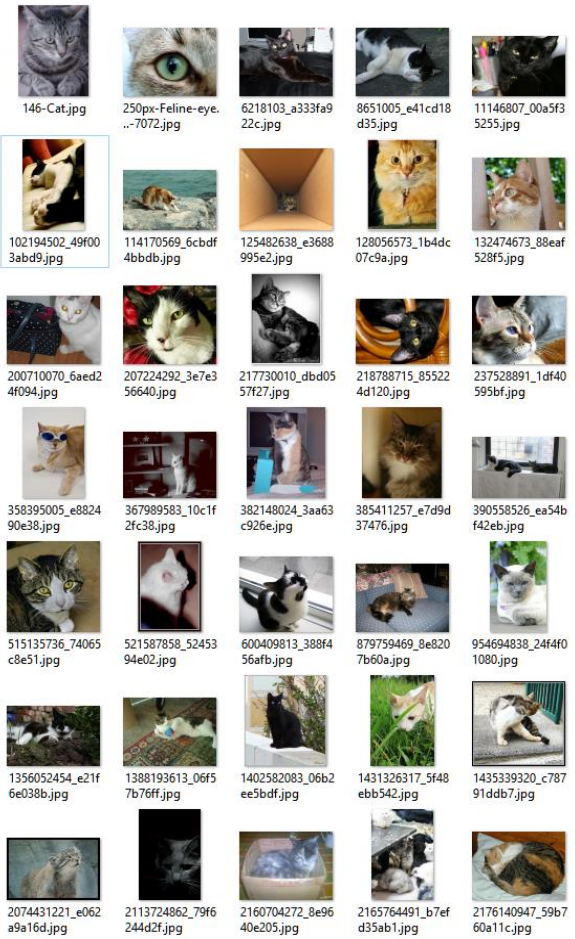
H_2O + TensorFlow
Live Demo

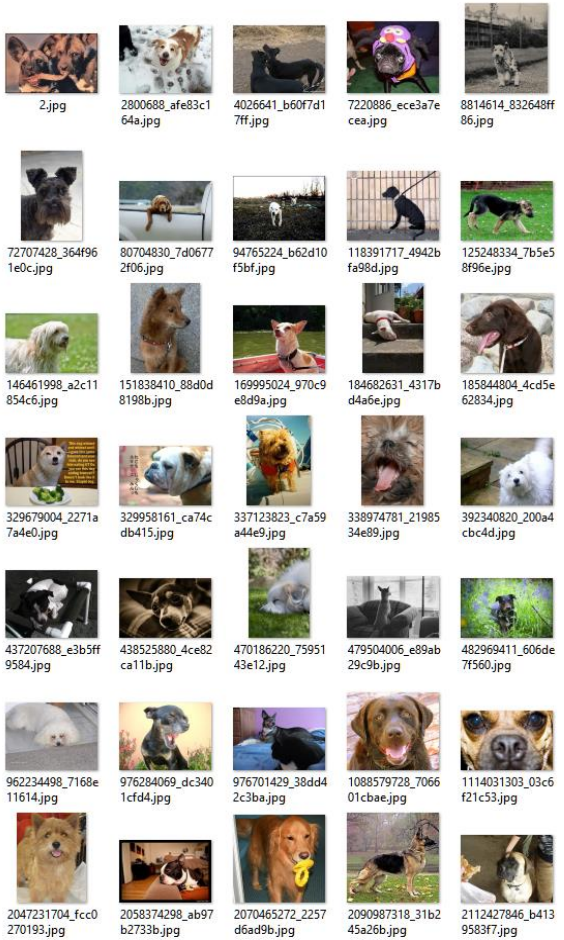
Deep Water H_2O + TensorFlow Demo

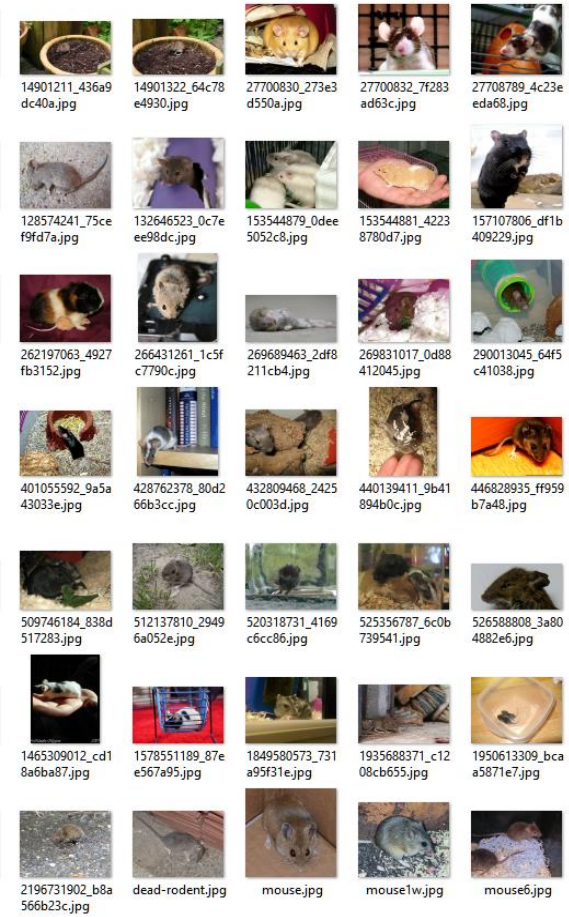
- H_2O + TensorFlow
  - Dataset – Cat/Dog/Mouse
  - TensorFlow as GPU backend
  - Train a LeNet (CNN) model
  - Interfaces
    - Python (Jupyter Notebook)
    - Web (H_2O Flow)

- Code and Data
  - github.com/h2oai/deepwater

Data – Cat/Dog/Mouse Images





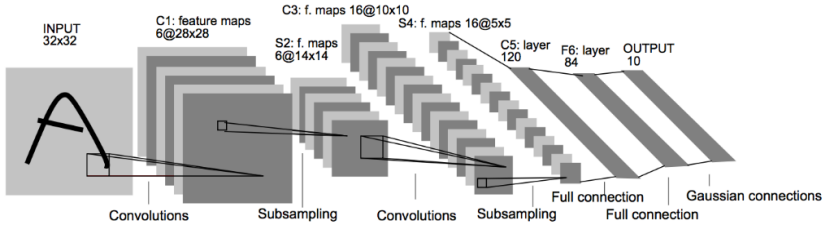


Data – CSV

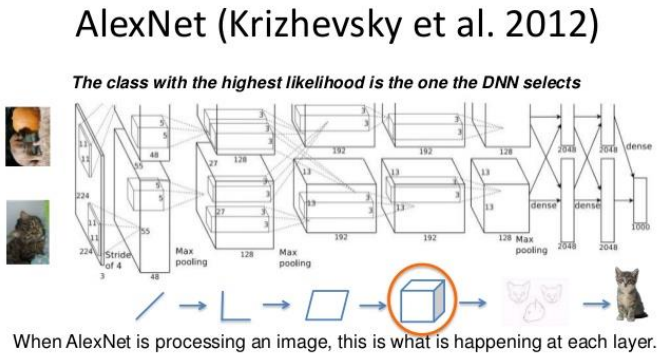
	A	B
1	bigdata/laptop/deepwater/imagenet/cat/102194502_49f003abd9.jpg	cat
2	bigdata/laptop/deepwater/imagenet/cat/11146807_00a5f35255.jpg	cat
3	bigdata/laptop/deepwater/imagenet/cat/1140846215_70e326f868.jpg	cat
4	bigdata/laptop/deepwater/imagenet/cat/114170569_6cbdf4bbdb.jpg	cat
5	bigdata/laptop/deepwater/imagenet/cat/1217664848_de4c7fc296.jpg	cat
6	bigdata/laptop/deepwater/imagenet/cat/1241603780_5e8c8f1ced.jpg	cat
7	bigdata/laptop/deepwater/imagenet/cat/1241612072_27ececbdef.jpg	cat
8	bigdata/laptop/deepwater/imagenet/cat/1241613138_ef1d82973f.jpg	cat
9	bigdata/laptop/deepwater/imagenet/cat/1244562192_35becd66bd.jpg	cat
10	bigdata/laptop/deepwater/imagenet/cat/125482638_e3688995e2.jpg	cat
11	bigdata/laptop/deepwater/imagenet/cat/128056573_1b4dc07c9a.jpg	cat
12	bigdata/laptop/deepwater/imagenet/cat/12945197_75e607e355.jpg	cat
13	bigdata/laptop/deepwater/imagenet/cat/132474673_88eaf528f5.jpg	cat
14	bigdata/laptop/deepwater/imagenet/cat/1350530984_ecf3039cf0.jpg	cat
15	bigdata/laptop/deepwater/imagenet/cat/1351606235_c9fbebf634.jpg	cat
16	bigdata/laptop/deepwater/imagenet/cat/1356052454_e21f6e038b.jpg	cat
17	bigdata/laptop/deepwater/imagenet/cat/1388193613_06f57b76ff.jpg	cat

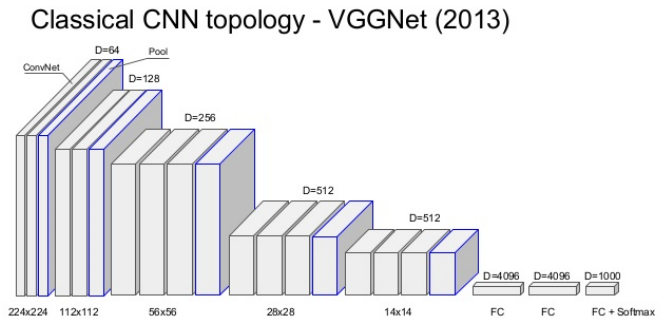
Available Networks in Deep Water

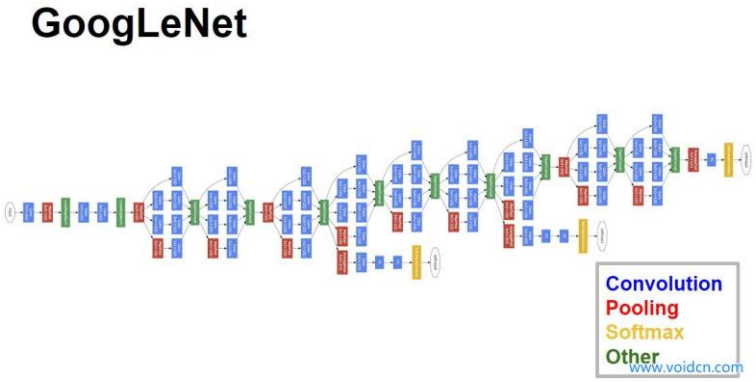
- LeNet (This Demo)
- AlexNet
- VGGNet
- Inception (GoogLeNet)
- ResNet (Deep Residual Learning)
- Build Your Own

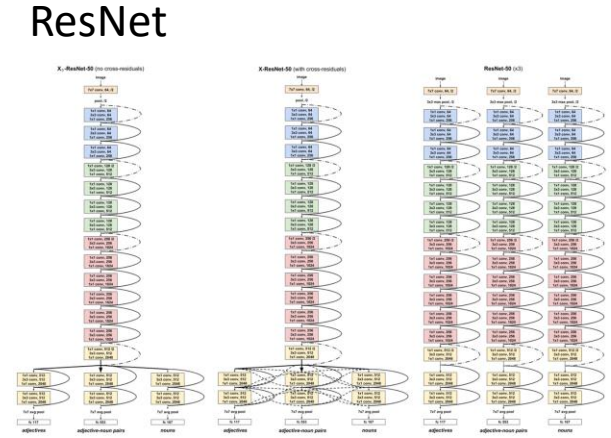


CNN called LeNet by Yann LeCun (1998)









Deep Water in Action

Quick Demo (5 minutes)

Using Tensorflow with H2O

This notebook shows how to use the tensorflow backend to tackle a simple image classification problem.

We start by connecting to our h2o cluster:

In [1]:
```
import h2o
h2o.init(port=54321, nthreads=-1)
```

```
Checking whether there is an H2O instance running at http://localhost:54321. connected.
```

H2O cluster uptime:	54 mins 37 secs
H2O cluster version:	3.11.0.99999
H2O cluster version age:	6 days
H2O cluster name:	ubuntu
H2O cluster total nodes:	1
H2O cluster free memory:	8.86 Gb
H2O cluster total cores:	8
H2O cluster allowed cores:	8
H2O cluster status:	locked, healthy
H2O connection url:	http://localhost:54321
H2O connection proxy:	None
Python version:	2.7.12 final

Then we make sure that the H2O cluster has the DeepWater distribution

In [2]:
```
from h2o.estimators.deepwater import H2ODeepWaterEstimator
if not H2ODeepWaterEstimator.available(): exit
```

H2O.ai

Load some python utilities library

In [3]:
```
import sys, os
import os.path
import pandas as pd
import numpy as np
import random
```

and finally we configure the IPython notebook to have nice visualizations

In [4]:
```
%matplotlib inline
from IPython.display import Image, display, HTML
import matplotlib.pyplot as plt
```

Configuration

Set the path to your h2o installation and download the 'bigdata' dataset using `./gradlew syncBigdataLaptop` from the H2O source distribution.

In [5]:
```
H2O_PATH=os.path.expanduser("~/h2o-3/")
```

Image Classification Task

H2O DeepWater allows you to specify a list of URIs (file paths) or URLs (links) to images, together with a response column (either a class membership (enum) or regression target (numeric)).

For this example, we use a small dataset that has a few hundred images, and three classes: cat, dog and mouse.

In [6]:
```
frame = h2o.import_file(H2O_PATH + "/bigdata/laptop/deepwater/imagenet/cat_dog_mouse.csv")
print(frame.dim)
print(frame.head(5))
```

```
Parse progress: |████████████████████████████████████████████████████████| 100%
[267, 2]
```

C1	C2
bigdata/laptop/deepwater/imagenet/cat/102194502_49f003abd9.jpg	cat
bigdata/laptop/deepwater/imagenet/cat/11146807_00a5f35255.jpg	cat
bigdata/laptop/deepwater/imagenet/cat/1140846215_70e326f868.jpg	cat
bigdata/laptop/deepwater/imagenet/cat/114170569_6cbdf4bbdb.jpg	cat
bigdata/laptop/deepwater/imagenet/cat/1217664848_de4c7fc296.jpg	cat

To build a LeNet image classification model in H2O, simply specify `network = "lenet"` and the **Tensorflow** backend to use the tensorflow lenet implementation:

In [12]:
```
model = H2ODeepWaterEstimator(epochs      = 500,
                              network     = "lenet",
                              image_shape = [28,28],  ## provide image size
                              channels    = 3,
                              backend     = "tensorflow",
                              model_id    = "deepwater_tf_simple")

model.train(x = [0], # file path e.g. xxx/xxx/xxx.jpg
            y = 1, # label cat/dog/mouse
            training_frame = frame)

model.show()
```

Home | (Busy) !TensorFlow_Par | 54.158.0.123/notebooks/!TensorFlow_Paris_Demo.ipynb

jupyter !TensorFlow_Paris_Demo Last Checkpoint: 13 minutes ago (u

File Edit View Insert Cell Kernel Help

Markdown CellToolbar

```
print(frame.head(5))
```

Parse progress: |

[267, 2]

C1
bigdata/laptop/deepwater/imagenet/cat/102194502_49f003abd9.jpg
bigdata/laptop/deepwater/imagenet/cat/11146807_00a5f35255.jpg
bigdata/laptop/deepwater/imagenet/cat/1140846215_70e326f868.jpg
bigdata/laptop/deepwater/imagenet/cat/114170569_6cbdf4bbdb.jpg
bigdata/laptop/deepwater/imagenet/cat/1217664848_de4c7fc296.jpg

ubuntu@ip-10-164-48-74: ~

ubuntu@ip-10-164-48-74: ~ 78x9

```
Every 2.0s: gpustat -cp                          Wed Nov 30 09:37:01 2016

ip-10-164-48-74  Wed Nov 30 09:37:01 2016
[0] GRID K520         | 34'C,  76 % |  3806 /  4036 MB | java/1357(3804M)
```

Using GPU for training

ubuntu@ip-10-164-48-74: ~ 78x9

```
  1  [|||||                                  15.2%]   5  [|||||                                  10.8%]
  2  [|||||||||                              26.2%]   6  [|||||                                  11.0%]
  3  [|||||                                  14.0%]   7  [|||||                                  10.5%]
  4  [|||||                                  11.1%]   8  [|||||                                  14.5%]
  Mem[|||||||||||||||||9.09G/14.7G]   Tasks: 50, 122 thr; 2 running
  Swp[                          0K/0K]   Load average: 0.33 0.21 0.15
                                         Uptime: 06:54:06
```

To build a LeNet image classification model in H2O, simply specify `network = "lenet"` and the **Tensorflow** backend to use the tensorflow lenet implementation:

In [*]:

```
model = H2ODeepWaterEstimator(epochs      = 500,
                              network     = "lenet",
                              image_shape = [28,28],  ## provide image size
                              channels    = 3,
                              backend     = "tensorflow",
                              model_id    = "deepwater_tf_simple")

model.train(x = [0], # file path e.g. xxx/xxx/xxx.jpg
            y = 1, # label cat/dog/mouse
            training_frame = frame)

model.show()
```

deepwater Model Build progress: |

H2O.ai

To build a LeNet image classification model in H2O, simply specify `network = "lenet"` and the **Tensorflow** backend to use the tensorflow lenet implementation:

In [12]:
```
model = H2ODeepWaterEstimator(epochs      = 500,
                              network     = "lenet",
                              image_shape = [28,28],  ## provide image size
                              channels    = 3,
                              backend     = "tensorflow",
                              model_id    = "deepwater_tf_simple")

model.train(x = [0], # file path e.g. xxx/xxx/xxx.jpg
            y = 1, # label cat/dog/mouse
            training_frame = frame)

model.show()
```

```
deepwater Model Build progress: |████████████████████████████████████████| 100%
Model Details
=============
H2ODeepWaterEstimator :  Deep Water
Model Key:  deepwater_tf_simple


ModelMetricsMultinomial: deepwater
** Reported on train data. **

MSE: 0.177650603738
RMSE: 0.421486184517
LogLoss: 0.865899719937
Mean Per-Class Error: 0.217708629345
Confusion Matrix: vertical: actual; across: predicted
```

cat	dog	mouse	Error	Rate
85.0	2.0	3.0	0.0555556	5 / 90
18.0	61.0	6.0	0.2823529	24 / 85
27.0	2.0	63.0	0.3152174	29 / 92
130.0	65.0	72.0	0.2172285	58 / 267

If you'd like to build your own Tensorflow network architecture, then this is easy as well. In this example script, we are using the **Tensorflow** backend. Models can easily be imported/exported between H2O and Tensorflow since H2O uses Tensorflow's format for model definition.

In [8]:
```
def simple_model(w, h, channels, classes):
    import json
    import tensorflow as tf
    # always create a new graph inside ipython or
    # the default one will be used and can lead to
    # unexpected behavior
    graph = tf.Graph()
    with graph.as_default():
        size = w * h * channels
        x = tf.placeholder(tf.float32, [None, size])
        W = tf.Variable(tf.zeros([size, classes]))
        b = tf.Variable(tf.zeros([classes]))
        y = tf.matmul(x, W) + b

        # labels
        y_ = tf.placeholder(tf.float32, [None, classes])

        # accuracy
        correct_prediction = tf.equal(tf.argmax(y, 1),
                                      tf.argmax(y_, 1))
        accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))

        # train
        cross_entropy = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(y, y_))
        train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)

        tf.add_to_collection("train", train_step)
        # this is required by the h2o tensorflow backend
        global_step = tf.Variable(0, name="global_step", trainable=False)

        init = tf.initialize_all_variables()
        tf.add_to_collection("init", init)
        tf.add_to_collection("logits", y)
        saver = tf.train.Saver()
        meta = json.dumps({
                "inputs": {"batch_image_input": x.name, "categorical_labels": y_.name},
                "outputs": {"categorical_logits": y.name},
                "metrics": {"accuracy": accuracy.name, "total_loss": cross_entropy.name},
                "parameters": {"global_step": global_step.name},
        })
        print(meta)
        tf.add_to_collection("meta", meta)
        filename = "/tmp/lenet_tensorflow.meta"
        tf.train.export_meta_graph(filename, saver_def=saver.as_saver_def())
    return filename
```

Saving the custom network structure as a file

H2O.ai

In [9]:
```
filename = simple_model(28, 28, 3, classes=3)
```


```
{"metrics": {"total_loss": "Mean_1:0", "accuracy": "Mean:0"}, "inputs": {"categorical_labels": "Placeholder_1:0",
 "batch_image_input": "Placeholder:0"}, "parameters": {"global_step": "global_step:0"}, "outputs": {"categorical_lo
gits": "add:0"}}
```


Creating the custom network structure with size = 28x28 and channels = 3

In [13]:
```
model = H2ODeepWaterEstimator(epochs                 = 500,
                              network_definition_file = filename,  ## specify the model
                              image_shape             = [28,28],   ## provide expected image size
                              channels                = 3,
                              backend                 = "tensorflow",
                              model_id                = "deepwater_tf_custom")

model.train(x = [0], # file path e.g. xxx/xxx/xxx.jpg
            y = 1, # label cat/dog/mouse
            training_frame = frame)

model.show()
```

Specifying the custom network structure for training

```
deepwater Model Build progress: |████████████████████████████████████████| 100%
Model Details
=============
H2ODeepWaterEstimator :  Deep Water
Model Key:  deepwater_tf_custom


ModelMetricsMultinomial: deepwater
** Reported on train data. **

MSE: 6.60075876885e+12
RMSE: 2569194.18668
LogLoss: -14.4921790248
Mean Per-Class Error: 0.0
Confusion Matrix: vertical: actual; across: predicted
```

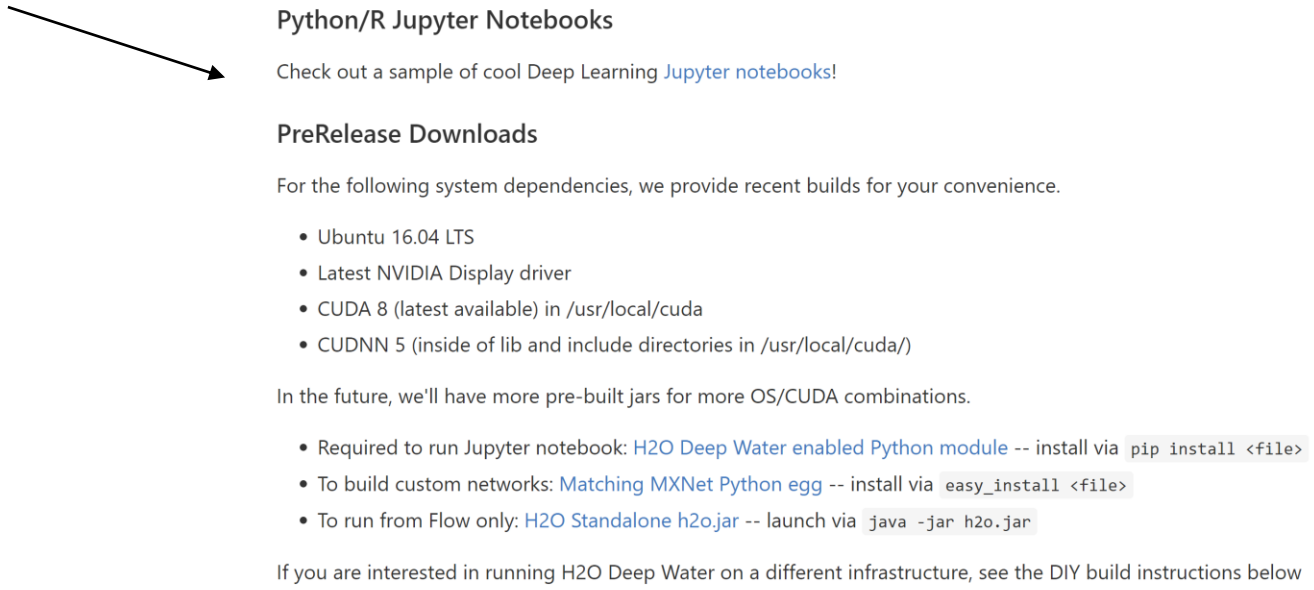
cat	dog	mouse	Error	Rate
90.0	0.0	0.0	0.0	0 / 90
0.0	85.0	0.0	0.0	0 / 85
0.0	0.0	92.0	0.0	0 / 92
90.0	85.0	92.0	0.0	0 / 267

H2O.ai

H_2O + TensorFlow
Now try it with H2O Flow

Want to try Deep Water?

- Build it
  - github.com/h2oai/deepwater
  - Ubuntu 16.04
  - CUDA 8
  - cuDNN 5
  - …
- Pre-built Amazon Machine Images (AMIs)
  - Info to be confirmed



Python/R Jupyter Notebooks

Check out a sample of cool Deep Learning Jupyter notebooks!

PreRelease Downloads

For the following system dependencies, we provide recent builds for your convenience.

- Ubuntu 16.04 LTS
- Latest NVIDIA Display driver
- CUDA 8 (latest available) in /usr/local/cuda
- CUDNN 5 (inside of lib and include directories in /usr/local/cuda/)

In the future, we'll have more pre-built jars for more OS/CUDA combinations.

- Required to run Jupyter notebook: H2O Deep Water enabled Python module -- install via `pip install <file>`
- To build custom networks: Matching MXNet Python egg -- install via `easy_install <file>`
- To run from Flow only: H2O Standalone h2o.jar -- launch via `java -jar h2o.jar`

If you are interested in running H2O Deep Water on a different infrastructure, see the DIY build instructions below

H_2O.ai Offers AI Open Source Platform

Product Suite to Operationalize Data Science with Visual Intelligence

Visual Intelligence and UX Framework For Data Interpretation and Story Telling on top of Beautiful Data Products

100% Open Source

H_2O.ai

In-Memory, Distributed Machine Learning Algorithms with Speed and Accuracy

Deep Water

State-of-the-art Deep Learning on GPUs with TensorFlow, MXNet or Caffe with the ease of use of H2O

Spark + H_2O

SPARKLING WATER

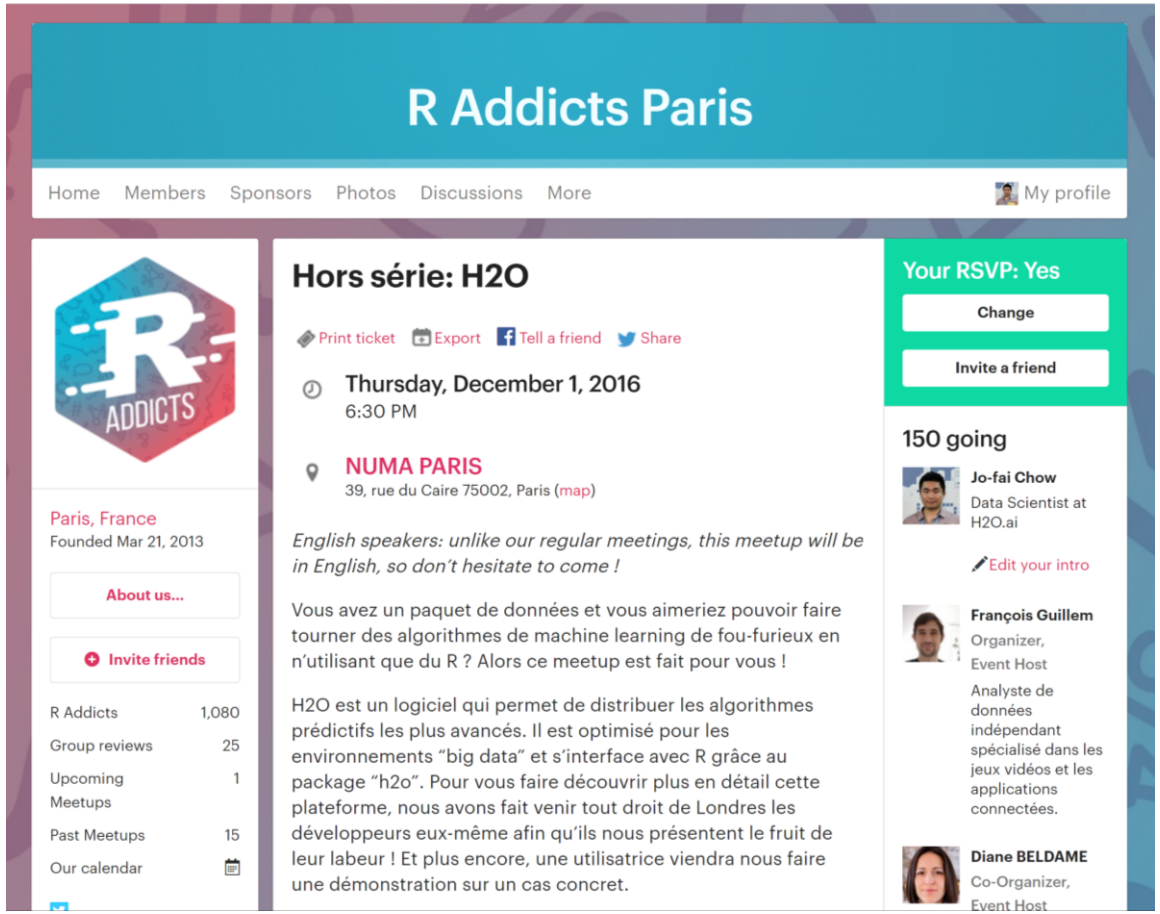
H2O Integration with Spark. Best Machine Learning on Spark.

Steam

Operationalize and Streamline Model Building, Training and Deployment Automatically and Elastically

Want to find out more about Sparkling Water and Steam?

- R Addicts Paris
- Tomorrow 6:30pm
- Three H_2O Talks:
  - Introduction to H_2O
    - Demos: H_2O + R + Steam
  - Auto Machine Learning using H_2O
  - Sparkling Water 2.0



Conclusions

Project “Deep Water”

- H_2O + TensorFlow
  - a powerful combination of two widely used open source machine learning libraries.
- All Goodies from H_2O
  - inherits all H_2O properties in scalability, ease of use and deployment.
- Unified Interface
  - allows users to build, stack and deploy deep learning models from different libraries efficiently.

- 100% Open Source
  - the party will get bigger!

H2O.ai

Deep Water Roadmap (Q4 2016)

Finish TensorFlow integration (C++/Python/Java):
Package Python on the backend to create trainable graphs

Finish Caffe integration (pure C++/Java):
Optimized Multi-GPU training (NVIDIA NCCL)

Add multi-GPU support for mxnet

Add more capabilities to H2O Deep Water:
Text/NLP, Time Series, LSTM, AutoEncoder,
Feature Extraction, Input/Output shape mapping, etc.

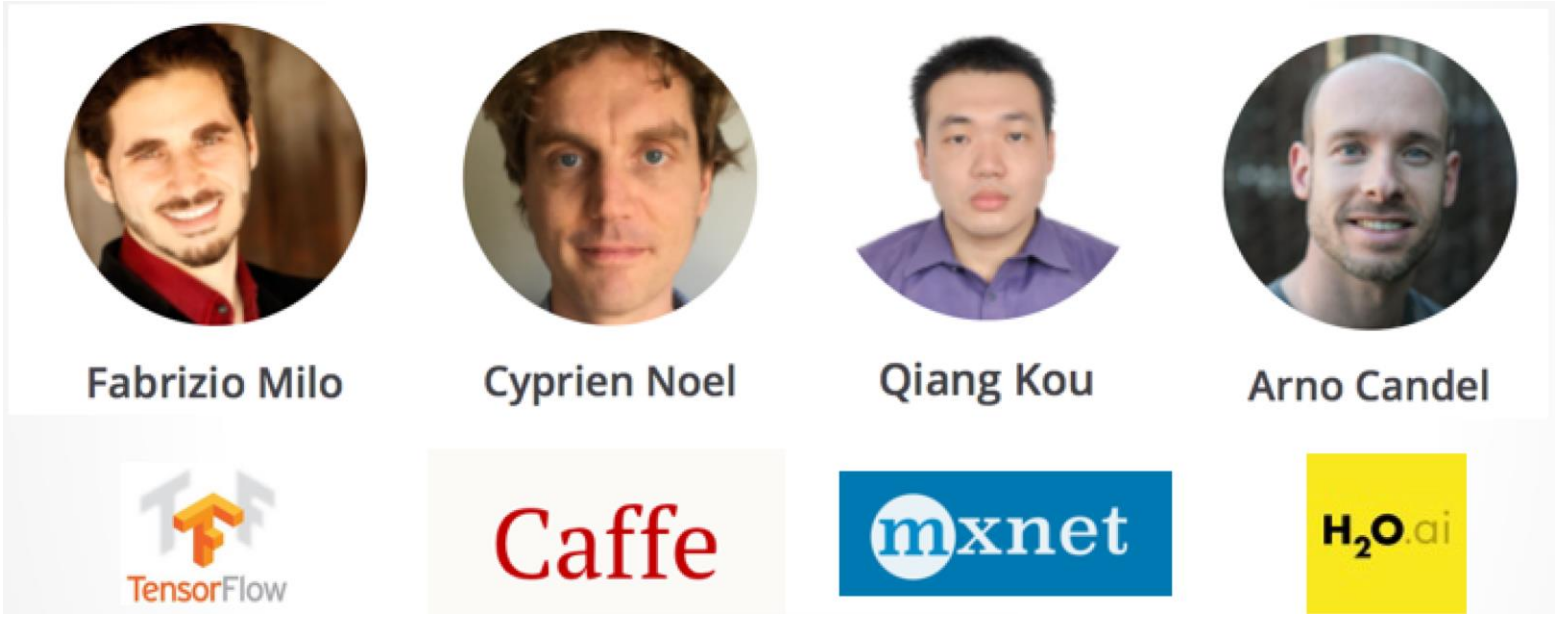
H2O.ai

H_2O's Mission

Making Machine Learning Accessible to Everyone

Photo credit: Virgin Media

Deep Water – Current Contributors



H2O.ai

Merci beaucoup!

- Organizers & Sponsors
  - Jiqiong, Natalia & Renat
  - Dailymotion
- Code, Slides & Documents
  - bit.ly/h2o_meetups
  - bit.ly/h2o_deepwater
  - docs.h2o.ai
- Contact
  - joe@h2o.ai
  - @matlabulous
  - github.com/woobe

TensorFlow™ PARIS

H2O.ai

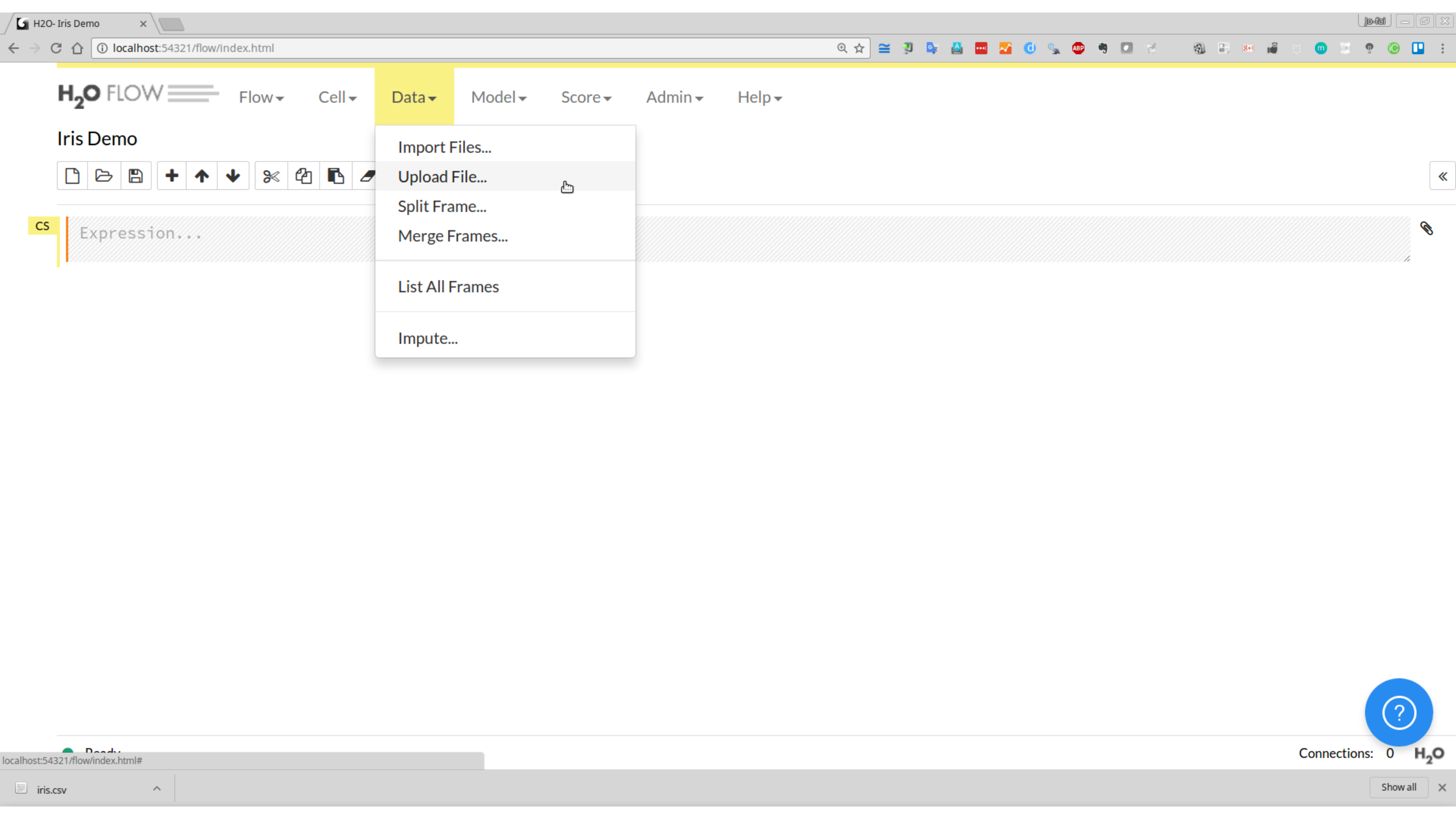
Merci beaucoup!

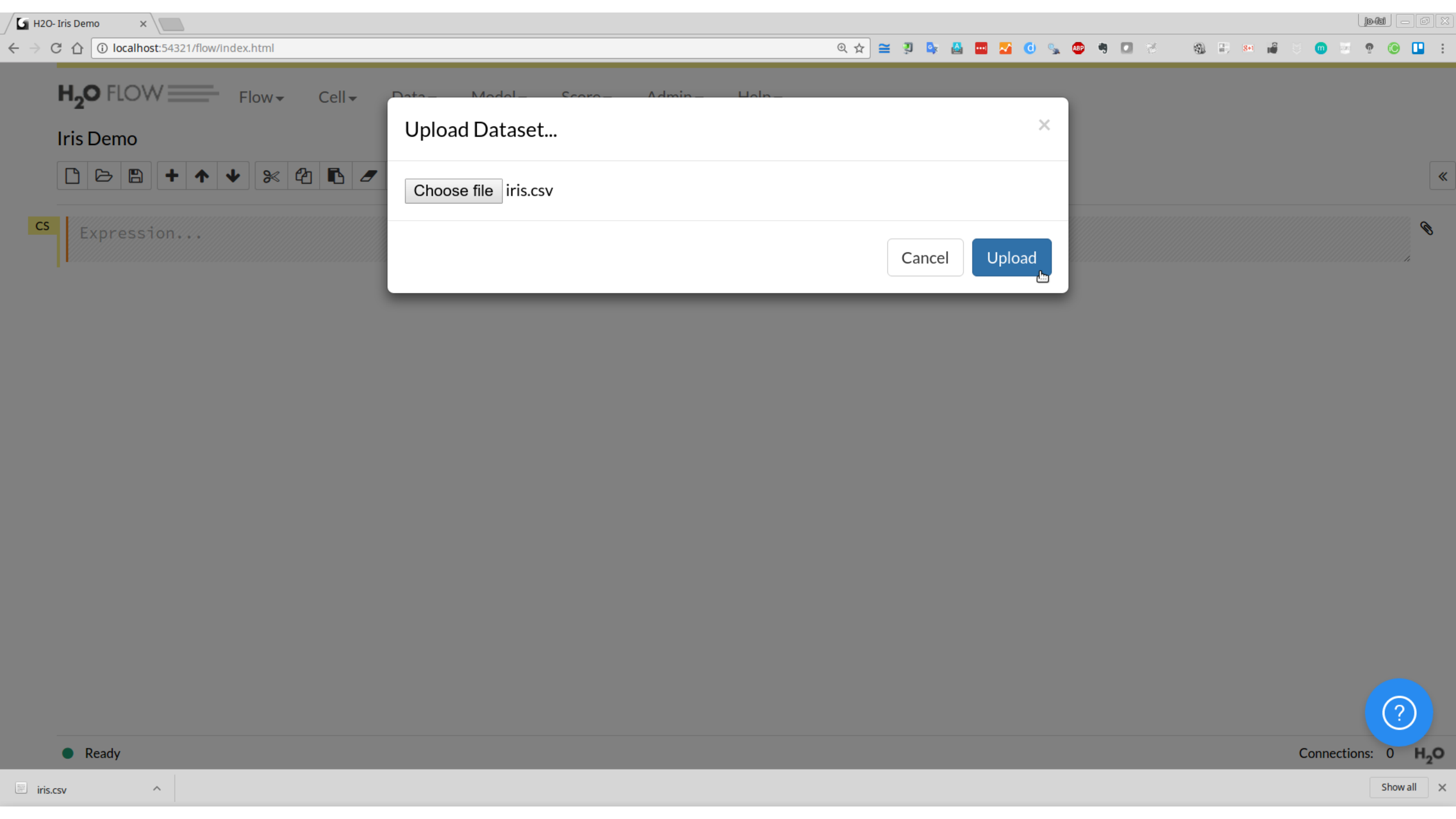
- Organizers & Sponsors
  - Jiqiong, Natalia & Renat
  - Dailymotion
- Code, Slides & Documents
  - bit.ly/h2o_meetups
  - bit.ly/h2o_deepwater
  - docs.h2o.ai
- Contact
  - joe@h2o.ai
  - @matlabulous
  - github.com/woobe

Haven't seen this before?

H2O.ai

Extra Slides (Iris Demo)


H2O- Iris Demo
localhost:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Iris Demo
Import Files...
Upload File...
Split Frame...
Merge Frames...
List All Frames
Impute...
CS
Expression...
Connections: 0
localhost:54321/flow/index.html#
iris.csv
Show all


H2O- Iris Demo
localhost:54321/flow/index.html
H2O FLOW
Flow
Cell
Iris Demo
CS
Expression...
Upload Dataset...
Choose file
iris.csv
Cancel
Upload
Ready
Connections: 0
iris.csv
Show all

Iris Demo

Setup Parse

PARSE CONFIGURATION

Sources iris.csv

ID iris_from_csv

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

Iris Demo

47ms

iris_from_csv

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

Rows	*Columns*	*Compressed Size*
150	5	1KB

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- Iris Demo

localhost:54321/flow/index.html#

H2O FLOW Flow Cell Data Model Score Admin Help

Iris Demo

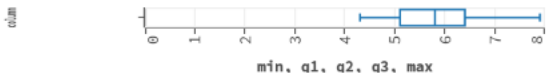
84ms

Summary: Sepal.Length

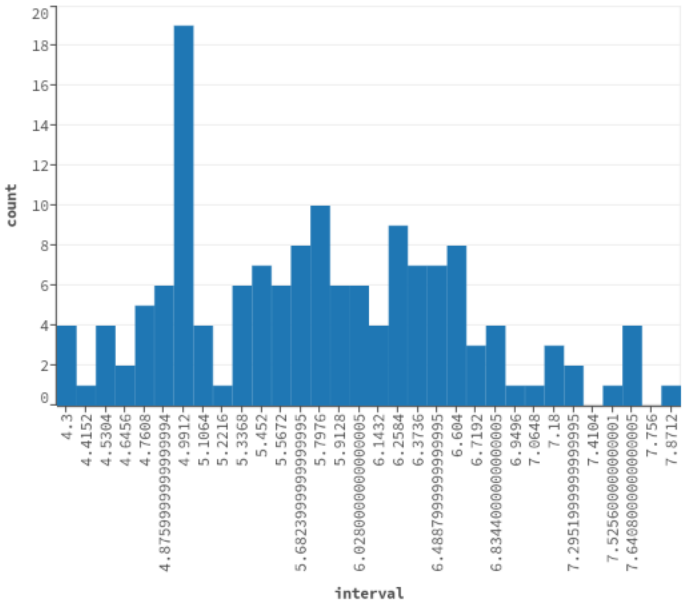
Actions Impute Inspect

CHARACTERISTICS

SUMMARY

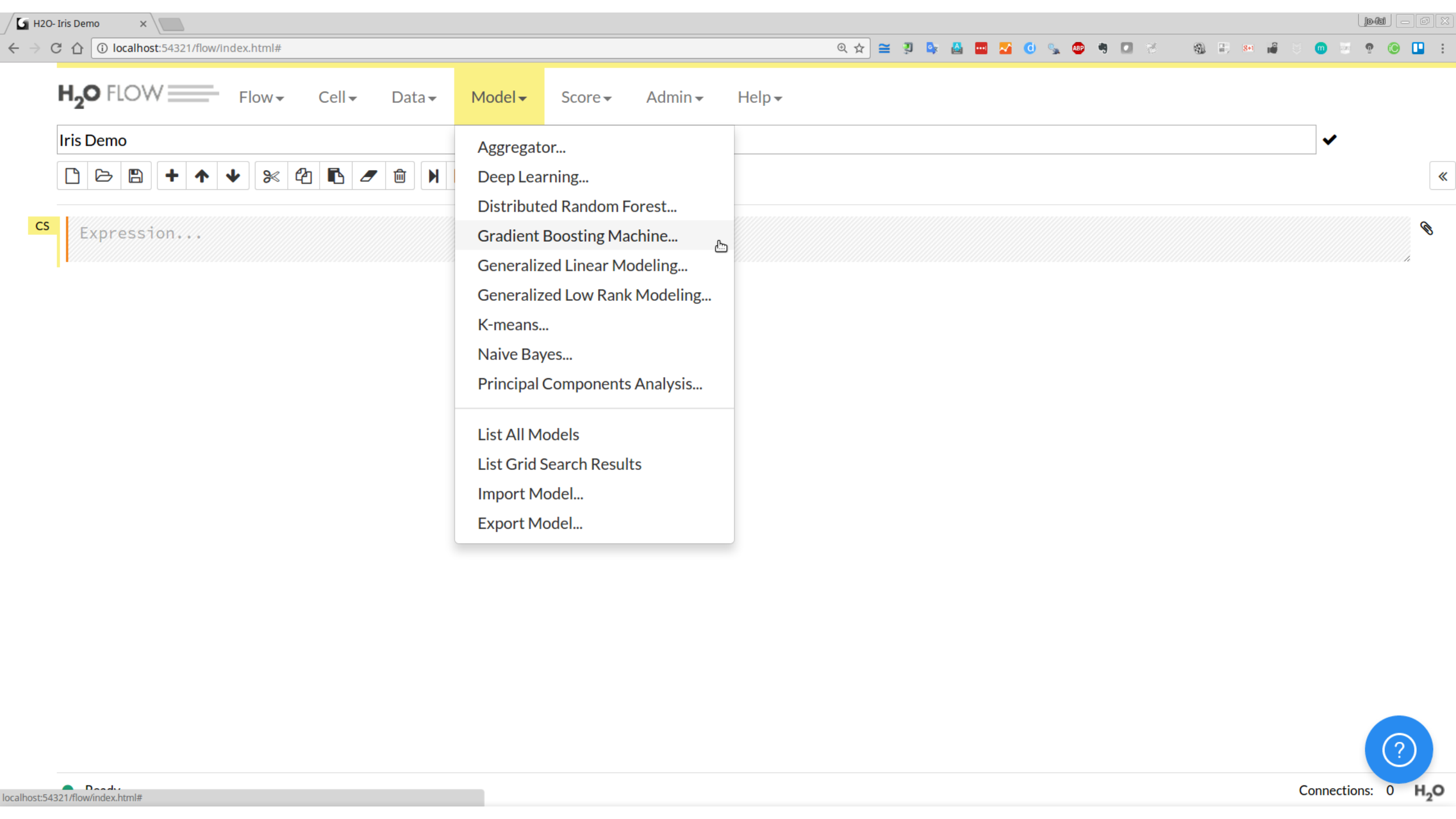


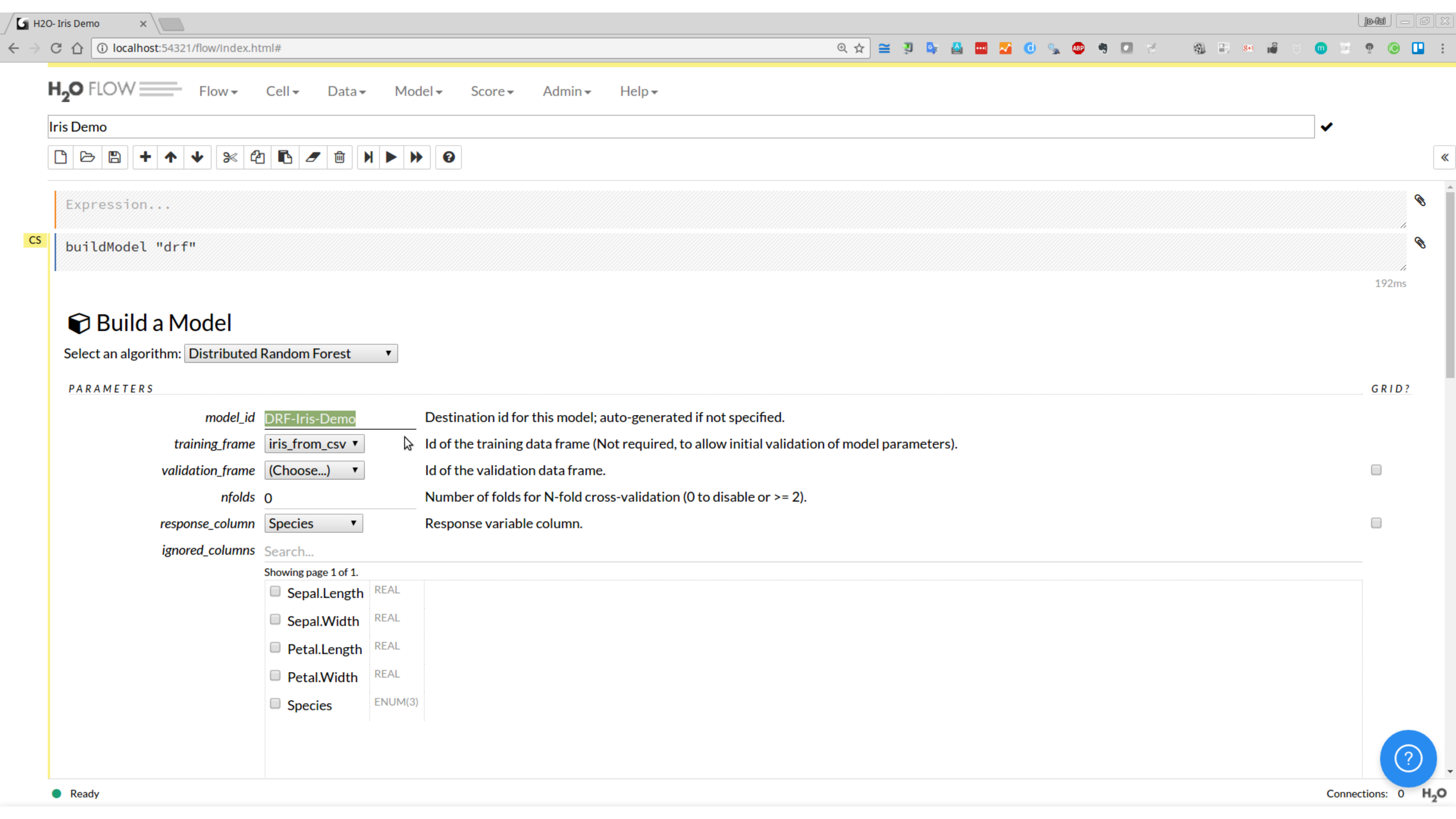
DISTRIBUTION

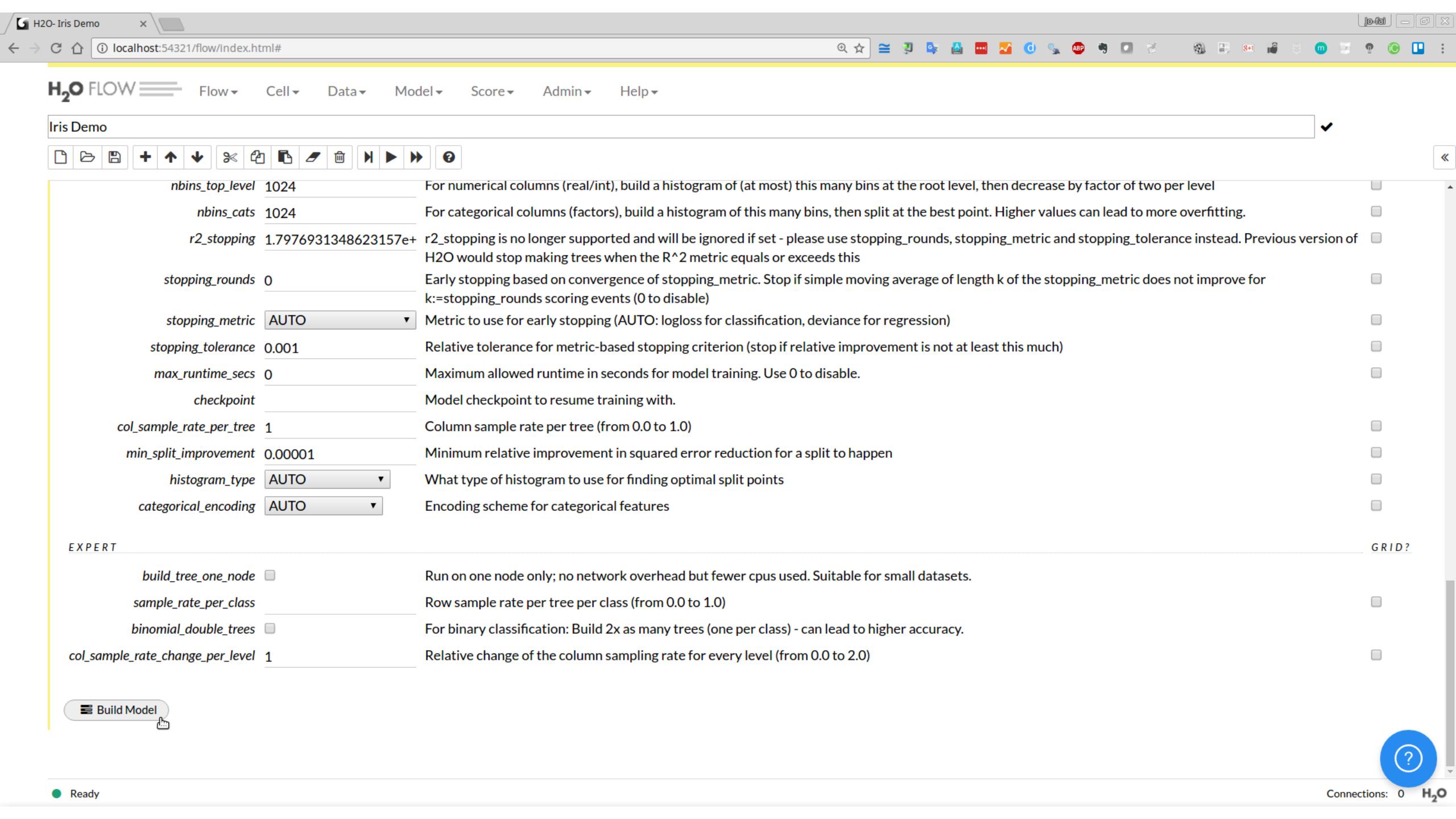


Ready

Connections: 0 H2O


H2O- Iris Demo
localhost:54321/flow/index.html#
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Iris Demo
CS
Expression...
Aggregator...
Deep Learning...
Distributed Random Forest...
Gradient Boosting Machine...
Generalized Linear Modeling...
Generalized Low Rank Modeling...
K-means...
Naive Bayes...
Principal Components Analysis...
List All Models
List Grid Search Results
Import Model...
Export Model...
Connections: 0
localhost:54321/flow/index.html#


H2O- Iris Demo
localhost:54321/flow/index.html#
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Iris Demo
Expression...
CS
buildModel "drf"
192ms
Build a Model
Select an algorithm: Distributed Random Forest
PARAMETERS
GRID?
model_id
DRF-Iris-Demo
Destination id for this model; auto-generated if not specified.
training_frame
iris_from_csv
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)
Ready
Connections: 0


H2O- Iris Demo
localhost:54321/flow/index.html#
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Iris Demo
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
Ready
Connections: 0

Iris Demo

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-Iris-Demo","training_frame":"iris_from_csv","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
,"score_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":1.79769
31348623157e+308,"stopping_rounds":0,"stopping_metric":"AUTO","stopping_tolerance":0.001,"max_runtime_secs":0,"checkpoint":"","col_sample_rate_per_tree":1,"min_sp
lit_improvement":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 00:00:00.183
Remaining Time 00:00:00.0
Type Model
Key DRF-Iris-Demo
Description DRF
Status DONE
Progress 100%

Done.

Actions View

Iris Demo

161ms

Model

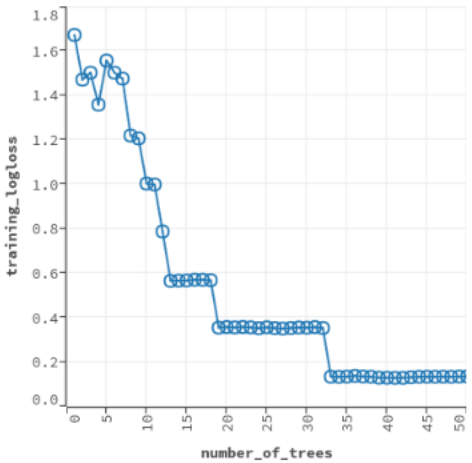
Model ID: DRF-Iris-Demo

Algorithm: Distributed Random Forest

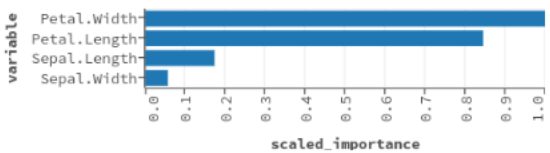
Actions: Refresh | Predict... | Download POJO | Download Model Deployment Package | Export | Inspect | Delete

MODEL PARAMETERS

SCORING HISTORY - LOGLOSS



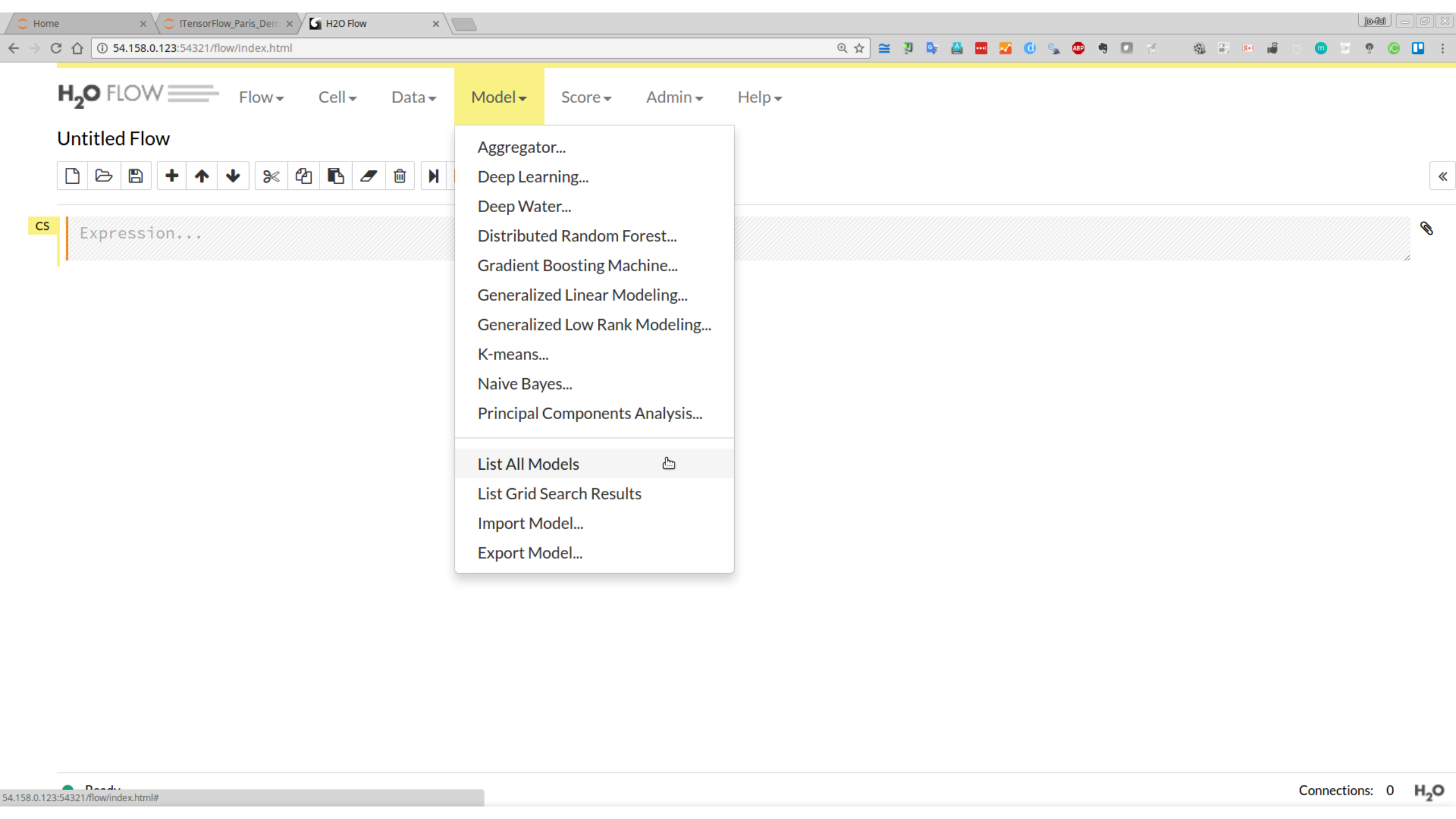
VARIABLE IMPORTANCES

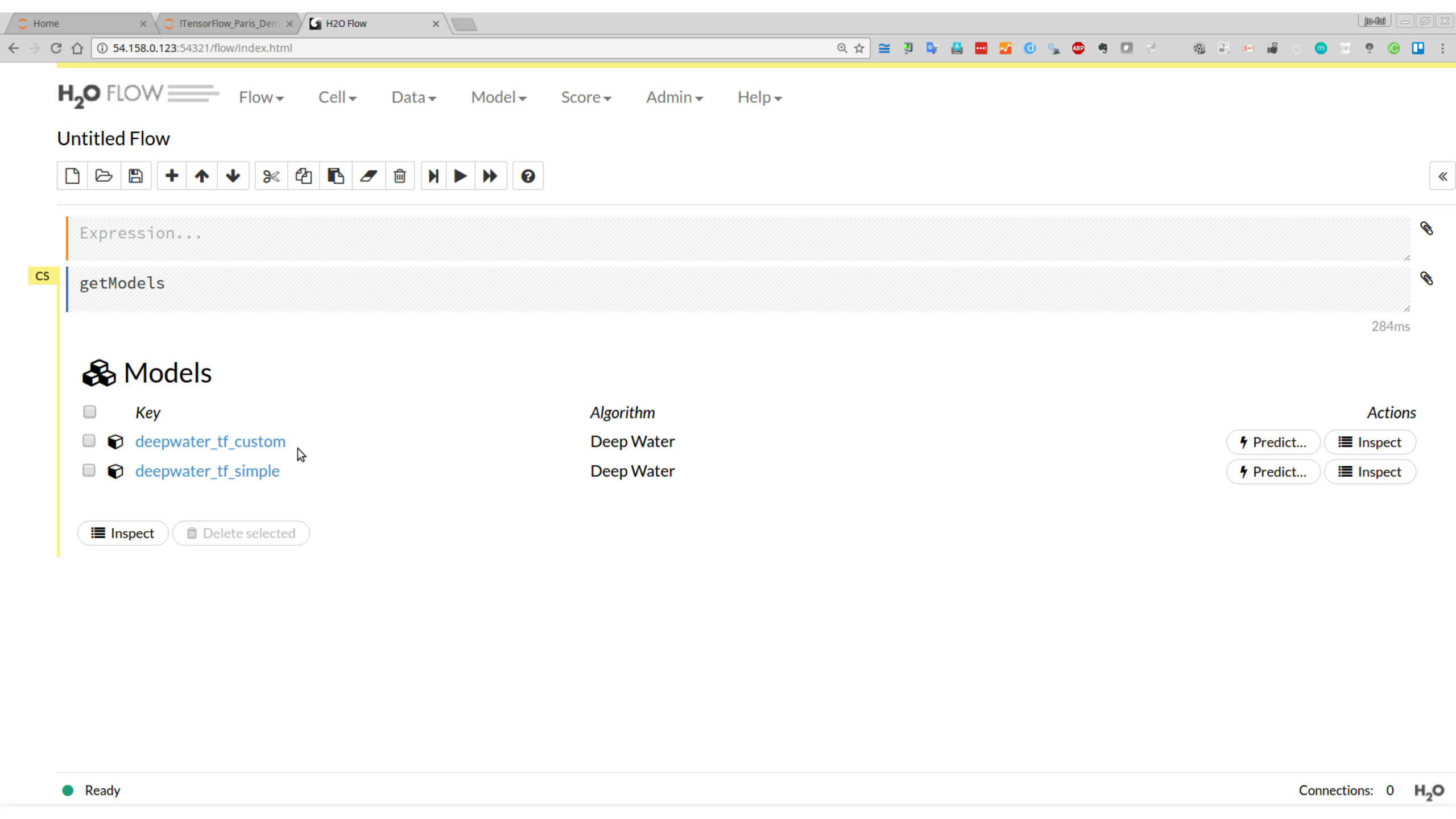


TRAINING METRICS - CONFUSION MATRIX VERTICAL: ACTUAL; ACROSS: PREDICTED

	setosa	versicolor	virginica	Error	Rate
setosa	50	0	0	**0**	**0 / 50**
versicolor	0	46	4	**0.0800**	**4 / 50**
virginica	0	4	46	**0.0800**	**4 / 50**
Total	**50**	**50**	**50**	**0.0533**	**8 / 150**

Extra Slides (Deep Water Demo)


Home
ITensorFlow_Paris_Dem
H2O Flow
54.158.0.123:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Untitled Flow
CS
Expression...
Aggregator...
Deep Learning...
Deep Water...
Distributed Random Forest...
Gradient Boosting Machine...
Generalized Linear Modeling...
Generalized Low Rank Modeling...
K-means...
Naive Bayes...
Principal Components Analysis...
List All Models
List Grid Search Results
Import Model...
Export Model...
Connections: 0
54.158.0.123:54321/flow/index.html#


Home
ITensorFlow_Paris_Dem
H2O Flow
54.158.0.123:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Untitled Flow
Expression...
CS
getModels
284ms
Models
Key
Algorithm
Actions
deepwater_tf_custom
Deep Water
Predict...
Inspect
deepwater_tf_simple
Deep Water
Predict...
Inspect
Inspect
Delete selected
Ready
Connections: 0

Model

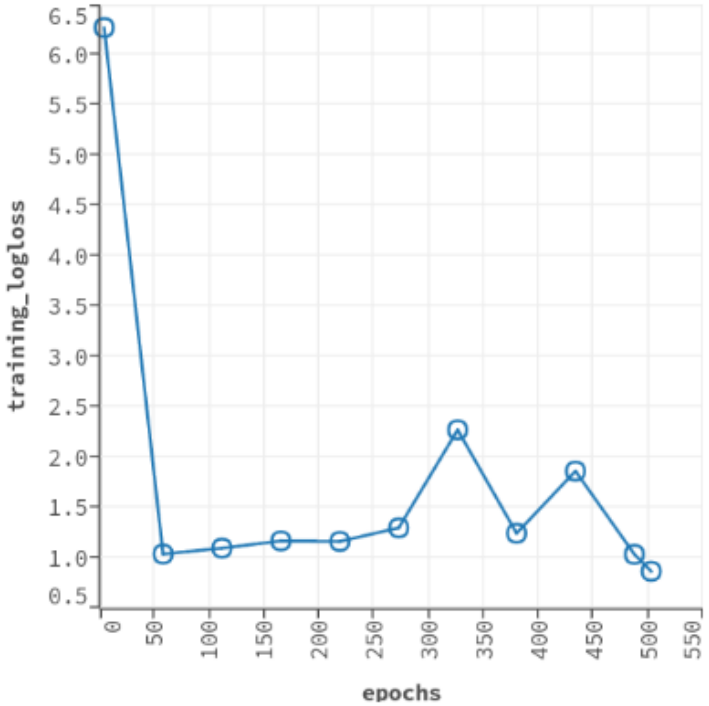
Model ID: deepwater_tf_simple

Algorithm: Deep Water

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

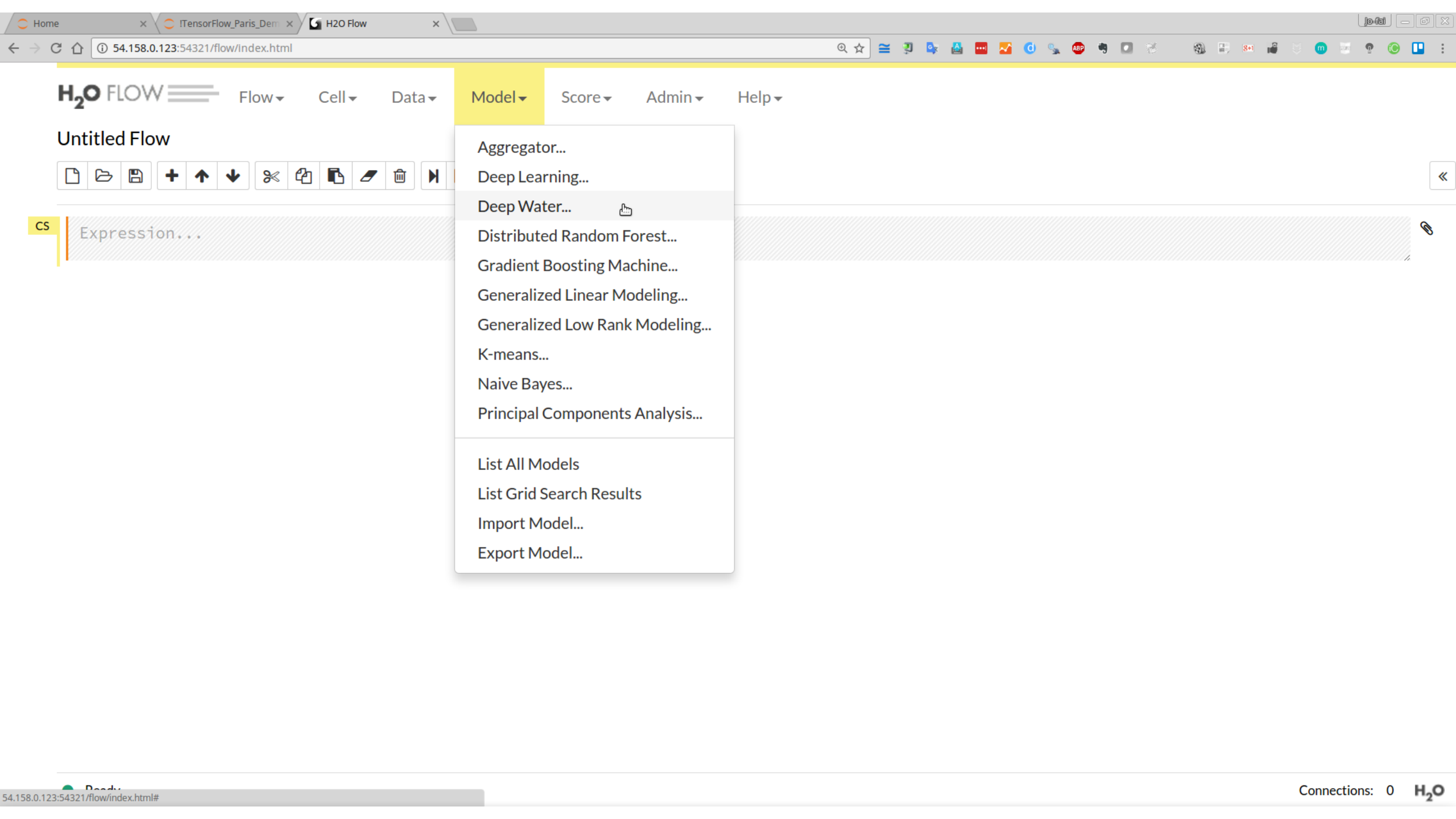
MODEL PARAMETERS

SCORING HISTORY - LOGLOSS



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

	cat	dog	mouse	Error	Rate
cat	85	2	3	**0.0556**	**5 / 90**
dog	18	61	6	**0.2824**	**24 / 85**
mouse	27	2	63	**0.3152**	**29 / 92**
Total	**130**	**65**	**72**	**0.2172**	**58 / 267**


Home
ITensorFlow_Paris_Dem
H2O Flow
54.158.0.123:54321/flow/index.html
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Untitled Flow
CS
Expression...
Aggregator...
Deep Learning...
Deep Water...
Distributed Random Forest...
Gradient Boosting Machine...
Generalized Linear Modeling...
Generalized Low Rank Modeling...
K-means...
Naive Bayes...
Principal Components Analysis...
List All Models
List Grid Search Results
Import Model...
Export Model...
Connections: 0
54.158.0.123:54321/flow/index.html#

Home | ITensorFlow_Paris_Dem | H2O- Deep Water H2O

54.158.0.123:54321/flow/index.html

H2O FLOW Flow Cell Data Model Score Admin Help

Deep Water H2O and TensorFlow Demo

Expression...

CS

```
buildModel "deepwater"
```

40ms

Build a Model

Select an algorithm: Deep Water

PARAMETERS GRID?

model_id	dw-tf-flow-demo	Destination id for this model; auto-generated if not specified.
training_frame	cat_dog_mouse.hex	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	C2	Response variable column.
ignored_columns	Search...	

Showing page 1 of 1.

C1	STRING
C2	ENUM(3)

Ready

Connections: 0

Home | ITensorFlow_Paris_Dem | H2O- Deep Water H2O

54.158.0.123:54321/flow/index.html

H2O FLOW Flow Cell Data Model Score Admin Help

Deep Water H2O and TensorFlow Demo

All None Previous 100 Next 100

Only show columns with more than 0 % missing values.

epochs	500	How many times the dataset should be iterated (streamed), can be fractional.
ignore_const_cols	☑	Ignore constant columns.
network	lenet	Network architecture.
activation		Activation function. Only used if no user-defined network architecture file is provided, and only for problem_type=dataset.
hidden		Hidden layer sizes (e.g. [200, 200]). Only used if no user-defined network architecture file is provided, and only for problem_type=dataset.
problem_type		Problem type, auto-detected by default. If set to image, the H2OFrame must contain a string column containing the path (URI or URL) to the images in the first column. If set to text, the H2OFrame must contain a string column containing the text in the first column. If set to dataset, Deep Water behaves just like any other H2O Model and builds a model on the provided H2OFrame (non-String columns).

(Choose...)
auto
user
lenet
alexnet
vgg
googlenet
inception_bn
resnet

ADVANCED GRID?

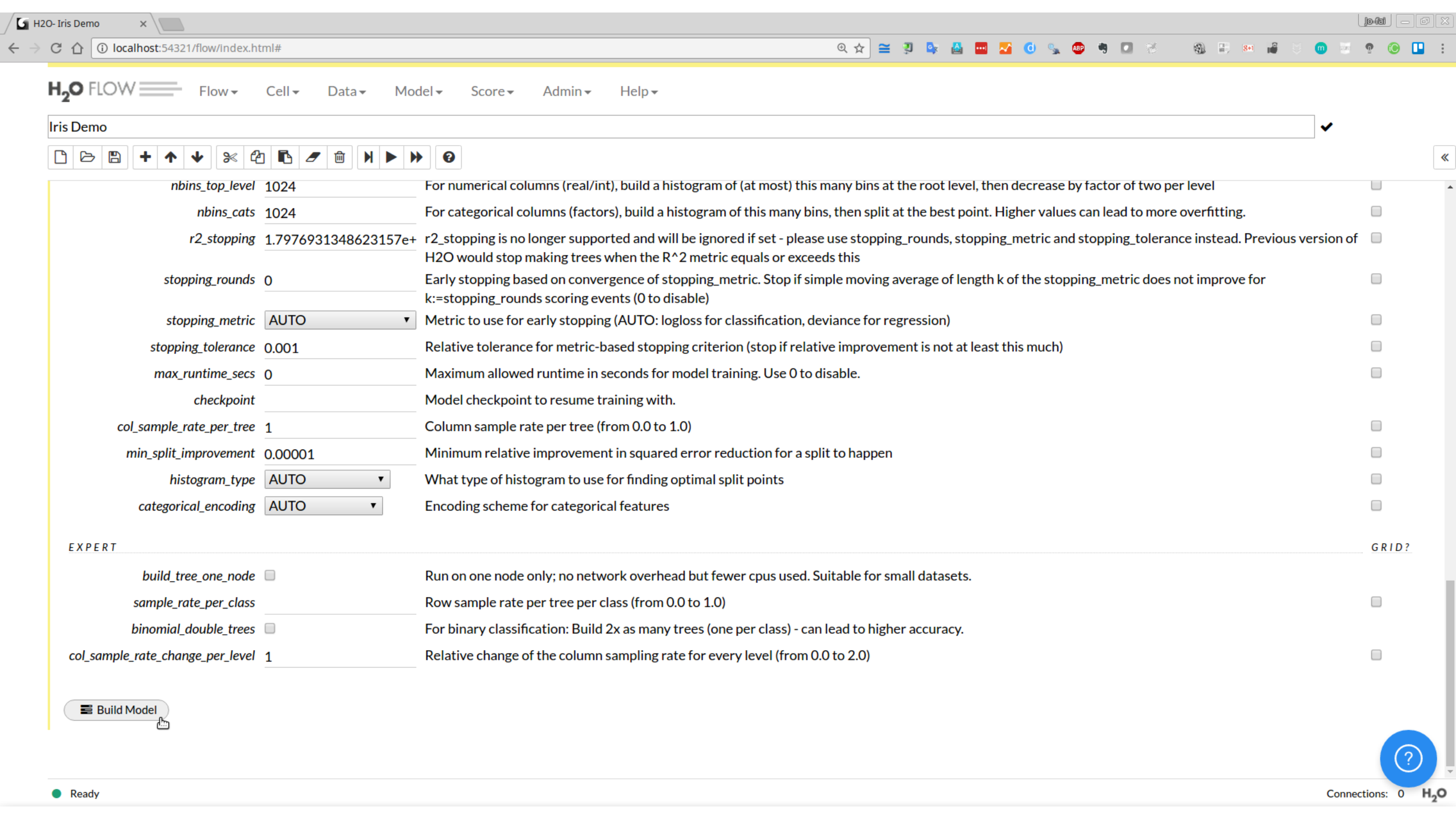
checkpoint		Model checkpoint to resume training with.
autoencoder	☐	Auto-Encoder.
balance_classes	☐	Balance training data class counts via over/under-sampling (for imbalanced data).
fold_column	(Choose...)	Column with cross-validation fold index assignment per observation.
offset_column	(Choose...)	Offset column. This will be added to the combination of columns before applying the link function.

Ready

Connections: 0 H2O

Deep Water H2O and TensorFlow Demo

score_training_samples	10000	Number of training set samples for scoring (0 for all).
score_validation_samples	0	Number of validation set samples for scoring (0 for all).
score_duty_cycle	1	Maximum duty cycle fraction for scoring (lower: more training, higher: more scoring).
stopping_rounds	5	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	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.
backend	tensorflow	Deep Learning Backend.
image_shape	28,28	Width and height of image.
channels	3	Number of (color) channels.
network_definition_file		Path of file containing network definition (graph, architecture).
network_parameters_file		Path of file containing network (initial) parameters (weights, biases).
mean_image_file		Path of file containing the mean image data for data normalization.
export_native_parameters_prefix		Path (prefix) where to export the native model parameters after every iteration.
input_dropout_ratio	0	Input layer dropout ratio (can improve generalization, try 0.1 or 0.2).
hidden_dropout_ratios		Hidden layer dropout ratios (can improve generalization), specify one value per hidden layer, defaults to 0.5.


H2O- Iris Demo
localhost:54321/flow/index.html#
H2O FLOW
Flow
Cell
Data
Model
Score
Admin
Help
Iris Demo
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
Ready
Connections: 0

Home | ITensorFlow_Paris_Dem | H2O- Deep Water H2O

54.158.0.123:54321/flow/index.html

H2O FLOW Flow Cell Data Model Score Admin Help

Deep Water H2O and TensorFlow Demo

CS

```
buildModel 'deepwater', {"model_id":"dw-tf-flow-demo","training_frame":"cat_dog_mouse.hex","nfolds":0,"response_column":"C2","ignored_columns":
[],"epochs":"500","ignore_const_cols":true,"network":"lenet","hidden":
[],"problem_type":"auto","checkpoint":"","autoencoder":false,"balance_classes":false,"score_each_iteration":false,"categorical_encoding":"AUTO","train_samples_per
_iteration":-2,"standardize":true,"distribution":"AUTO","score_interval":5,"score_training_samples":10000,"score_validation_samples":0,"score_duty_cycle":"1","sto
pping_rounds":5,"stopping_metric":"AUTO","stopping_tolerance":0,"max_runtime_secs":0,"backend":"tensorflow","image_shape":
[28,28],"channels":3,"network_definition_file":"","network_parameters_file":"","mean_image_file":"","export_native_parameters_prefix":"","input_dropout_ratio":0,"
hidden_dropout_ratios":
[],"overwrite_with_best_model":true,"target_ratio_comm_to_comp":0.05,"seed":-1,"learning_rate":0.005,"learning_rate_annealing":0.000001,"momentum_start":0.9,"mome
ntum_ramp":10000,"momentum_stable":0.99,"shuffle_training_data":true,"mini_batch_size":32,"clip_gradient":10,"sparse":false,"gpu":true,"device_id":[0]}
```

Started at 10:34:21 am

Job

Run Time 00:00:17.189

Remaining Time 00:00:37.468

Type Model

Key dw-tf-flow-demo

Description DeepWater

Status RUNNING

Progress 32%

Iterations: 41. Epochs: 157.243. Speed: 2505 samples/sec. Estimated time left: 38.387 sec

Actions View | Cancel Job

Requesting /3/Jobs/%2403010aa4304a32d4ffffffff%24_80bfd4d20d78dd19c3f2b2eb01544f85

Connections: 1

Home | TensorFlow_Paris_Dem | H2O- Deep Water H2O

54.158.0.123:54321/flow/index.html

H2O FLOW Flow Cell Data Model Score Admin Help

Deep Water H2O and TensorFlow Demo

CS

```
buildModel 'deepwater', {"model_id":"dw-tf-flow-demo","training_frame":"cat_dog_m
[],"epochs":"500","ignore_const_cols":true,"network":"lenet","hidden":
[],"problem_type":"auto","checkpoint":"","autoencoder":false,"balance_classes":fa
_iteration":-2,"standardize":true,"distribution":"AUTO","score_interval":5,"score
pping_rounds":5,"stopping_metric":"AUTO","stopping_tolerance":0,"max_runtime_secs
[28,28],"channels":3,"network_definition_file":"","network_parameters_file":"","m
hidden_dropout_ratios":
[],"overwrite_with_best_model":true,"target_ratio_comm_to_comp":0.05,"seed":-1,"l
ntum_ramp":10000,"momentum_stable":0.99,"shuffle_training_data":true,"mini_batch_
```

Started at 10:34:21 am

Job

Run Time	00:00:34.471
Remaining Time	00:00:17.785
Type	Model
Key	dw-tf-flow-demo
Description	DeepWater
Status	RUNNING
Progress	66%

Iterations: 86. Epochs: 329.828. Speed: 2585 samples/sec. Estimated time left: 18.361 sec

Actions View | Cancel Job

ubuntu@ip-10-164-48-74: ~

ubuntu@ip-10-164-48-74: ~ 78x9

```
Every 2.0s: gpustat -cp                                  Wed Nov 30 09:34:54 2016

ip-10-164-48-74  Wed Nov 30 09:34:54 2016
[0] GRID K520         | 36'C,  77 % |  3806 /  4036 MB | java/1357(3804M)
```

ubuntu@ip-10-164-48-74: ~ 78x9

```
  1  [||||                                  10.1%]   5  [|||||||                                 23.8%]
  2  [|||||                                 12.0%]   6  [|||                                     10.3%]
  3  [||||||                                14.6%]   7  [||||                                    11.9%]
  4  [||||                                  11.3%]   8  [|||                                      9.6%]
  Mem[||||||||||||||||||8.64G/14.7G]   Tasks: 50, 120 thr; 2 running
  Swp[                    0K/0K]   Load average: 0.37 0.16 0.12
                                   Uptime: 06:51:59
```

Ready

Connections: 0

ubuntu@ip-10-164-48-74: ~ | H2O- Deep Water H2O and...

Wed, Nov 30, 2016 10:34 am

Model

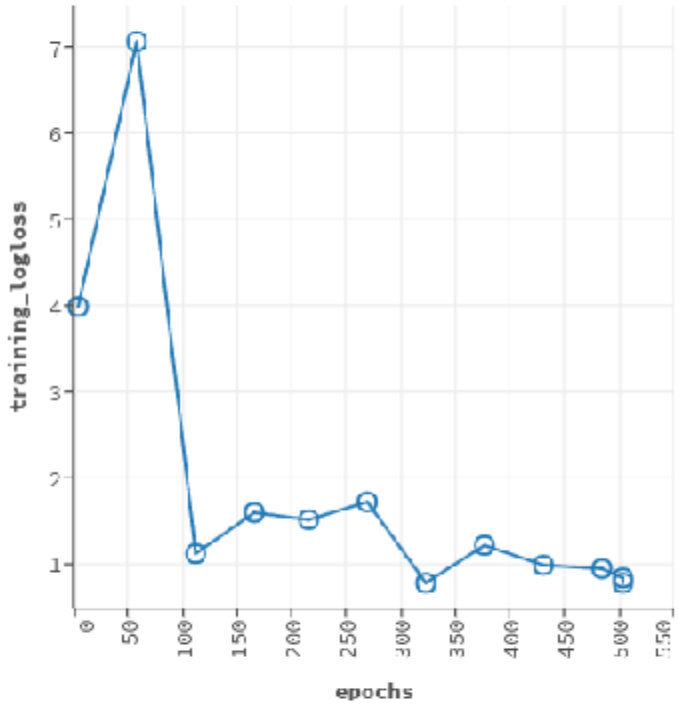
Model ID: dw-tf-flow-demo

Algorithm: Deep Water

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

MODEL PARAMETERS

SCORING HISTORY - LOGLOSS



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

	cat	dog	mouse	Error	Rate
cat	83	5	2	**0.0778**	**7 / 90**
dog	4	78	3	**0.0824**	**7 / 85**
mouse	14	10	68	**0.2609**	**24 / 92**
Total	**101**	**93**	**73**	**0.1423**	**38 / 267**