Introduction to H2O

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

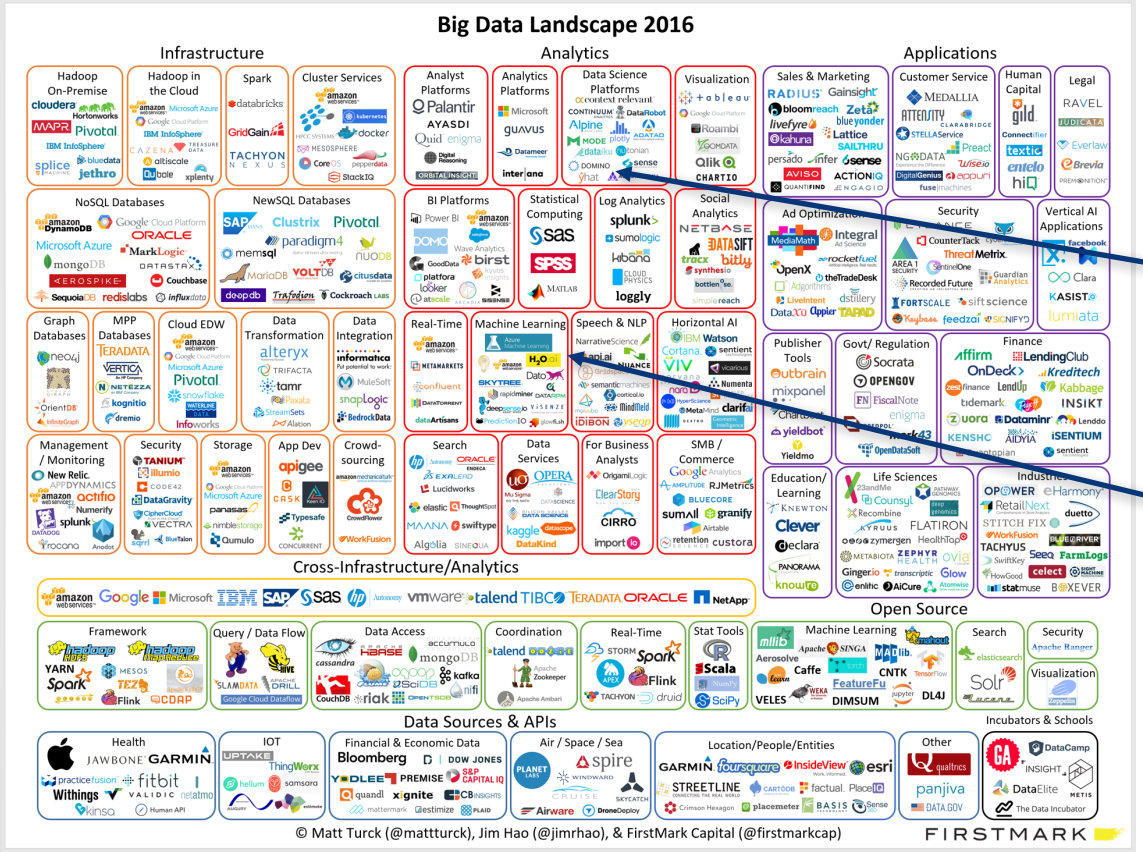
Data Science for IoT Meetup – London
July 2016

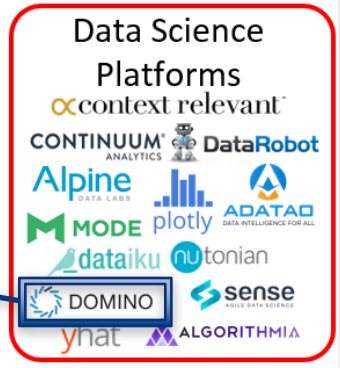
About Me: Engineer → Data Scientist

- 2005 - 2015
- Water Engineer
  - Consultant for Utilities
    - SEAMS (Sheffield)
  - EngD Research
    - Water + Machine Learning
    - University of Exeter
    - XP Solutions (Newbury)

- 2015 - Present
- Data Scientist
  - UK Telecom
    - Virgin Media
  - Silicon Valley
    - Domino Data Lab
    - H2O.ai

My Silicon Valley Journey





Agenda

- Who/What is H2O?
- H2O Machine Learning Platform
- H2O in Web, R and Python
- H2O Use Cases
- What's Next?

ABOUT H2O

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

H2O.ai, the Company

- Team: 55; Founded in 2012
- Mountain View, CA
- Stanford & Purdue Math & Systems Engineers

H2O, the Platform

- Open Source Software (Apache 2.0 Licensed)
- R, Python, Scala, Java and Web Interfaces
- Distributed Algorithms that Scale to Big Data

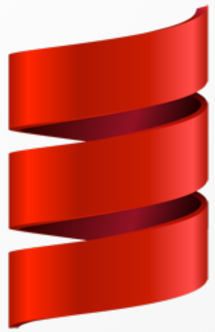
H2O

H2O PLATFORM

H2O Platform Overview

- Distributed implementations of cutting edge ML algorithms.
- Core algorithms written in high performance Java.
- APIs available in R, Python, Scala, REST/JSON.
- Interactive Web GUI.




H2O Platform Overview

- Write code in high-level language like R (or use the web GUI) and output production-ready models in Java.
- To scale, just add nodes to your H2O cluster.
- Works with Hadoop, Spark and your laptop.

H2O Distributed Computing

H2O Cluster

- Multi-node cluster with shared memory model.
- All computations in memory.
- Each node sees only some rows of the data.
- No limit on cluster size.

Distributed K/V Store

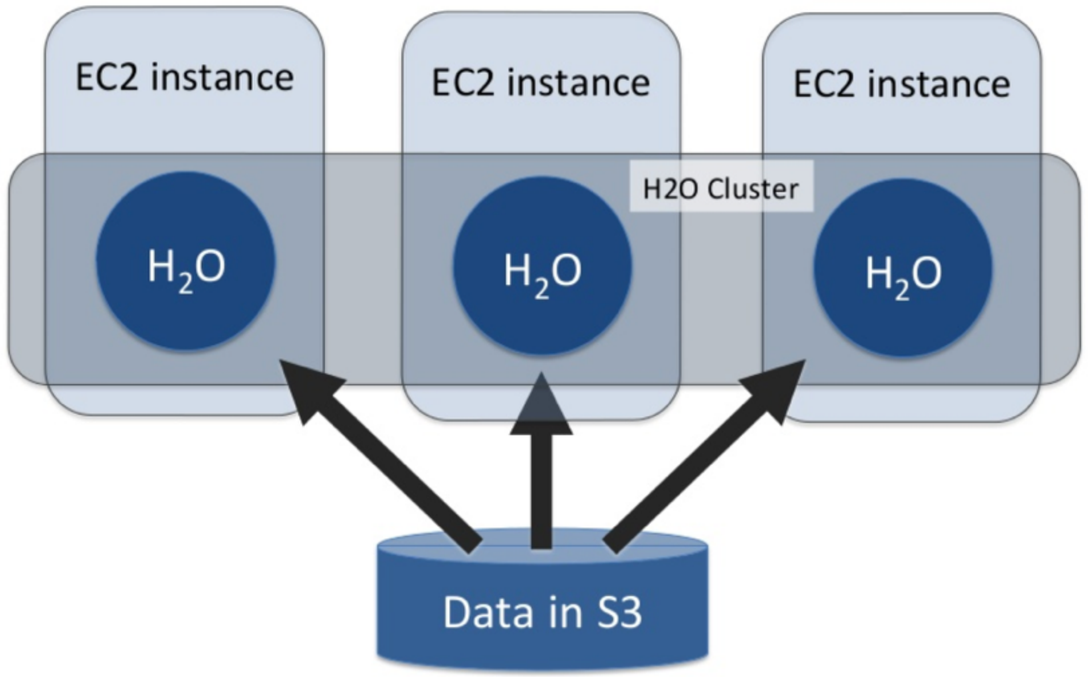
- Objects in the H2O cluster such as data frames, models and results are all referenced by key.
- Any node in the cluster can access any object in the cluster by key.

H2O Frame

- Distributed data frames (collection of vectors).
- Columns are distributed (across nodes) arrays.
- Works just like R's data.frame or Python Pandas DataFrame

H2O

H2O on Amazon EC2



H2O can easily be deployed on an Amazon EC2 cluster.

The H2O GitHub repository contains example scripts that help to automate the cluster deployment ("ec2" folder).

H2O API FOR R / PYTHON

h2o R Package

Installation

- Java 7 or later; R 3.1 and above; Linux, Mac, Windows
- The easiest way to install the h2o R package is CRAN.
- Latest version: http://www.h2o.ai/download/h2o/r

Design

All computations are performed in highly optimized Java code in the H2O cluster, initiated by REST calls from R.

h2o Python Module

Installation

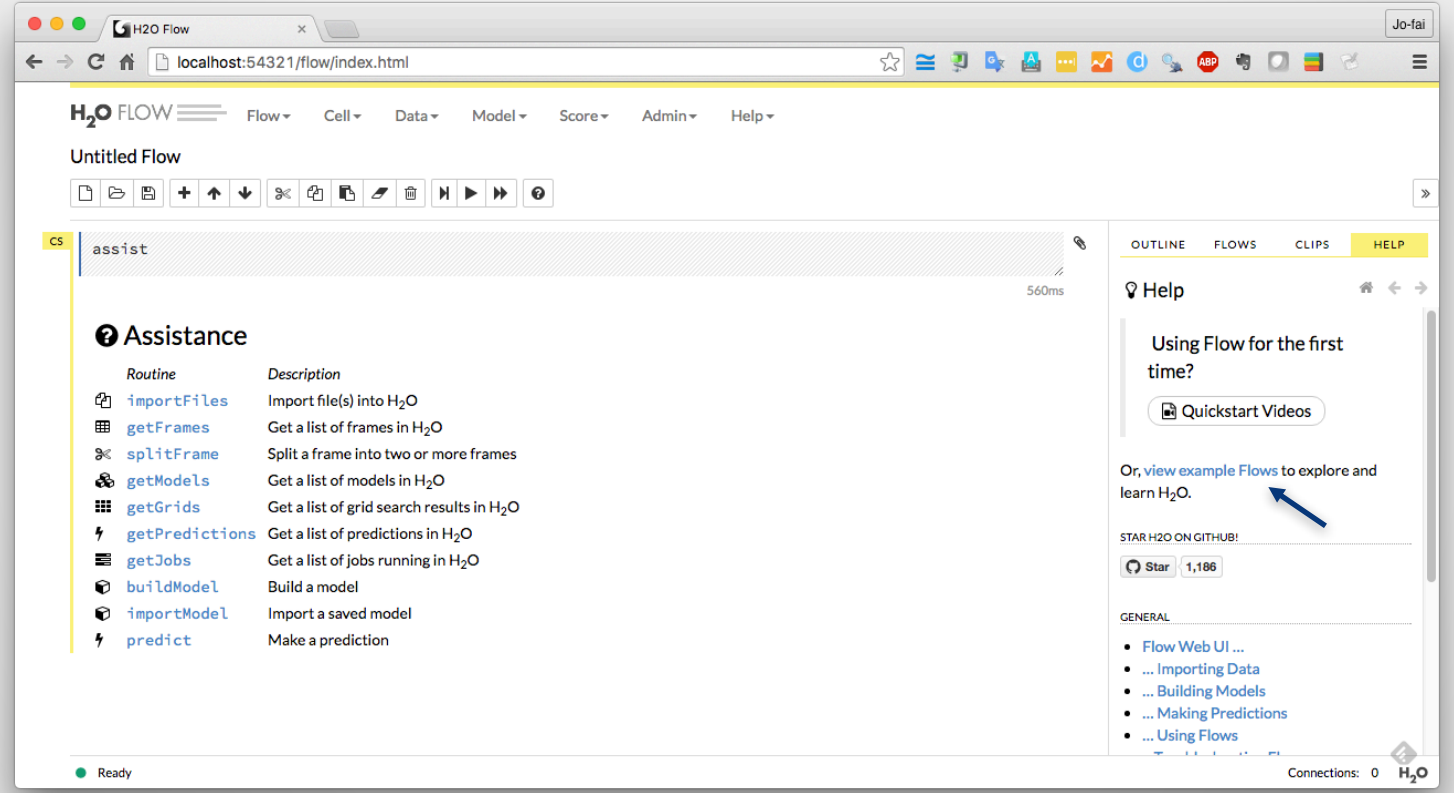
- Java 7 or later; Python 2.7, 3.5; Linux, Mac, Windows
- The easiest way to install the h2o Python module is PyPI.
- Latest version: http://www.h2o.ai/download/h2o/python

Design

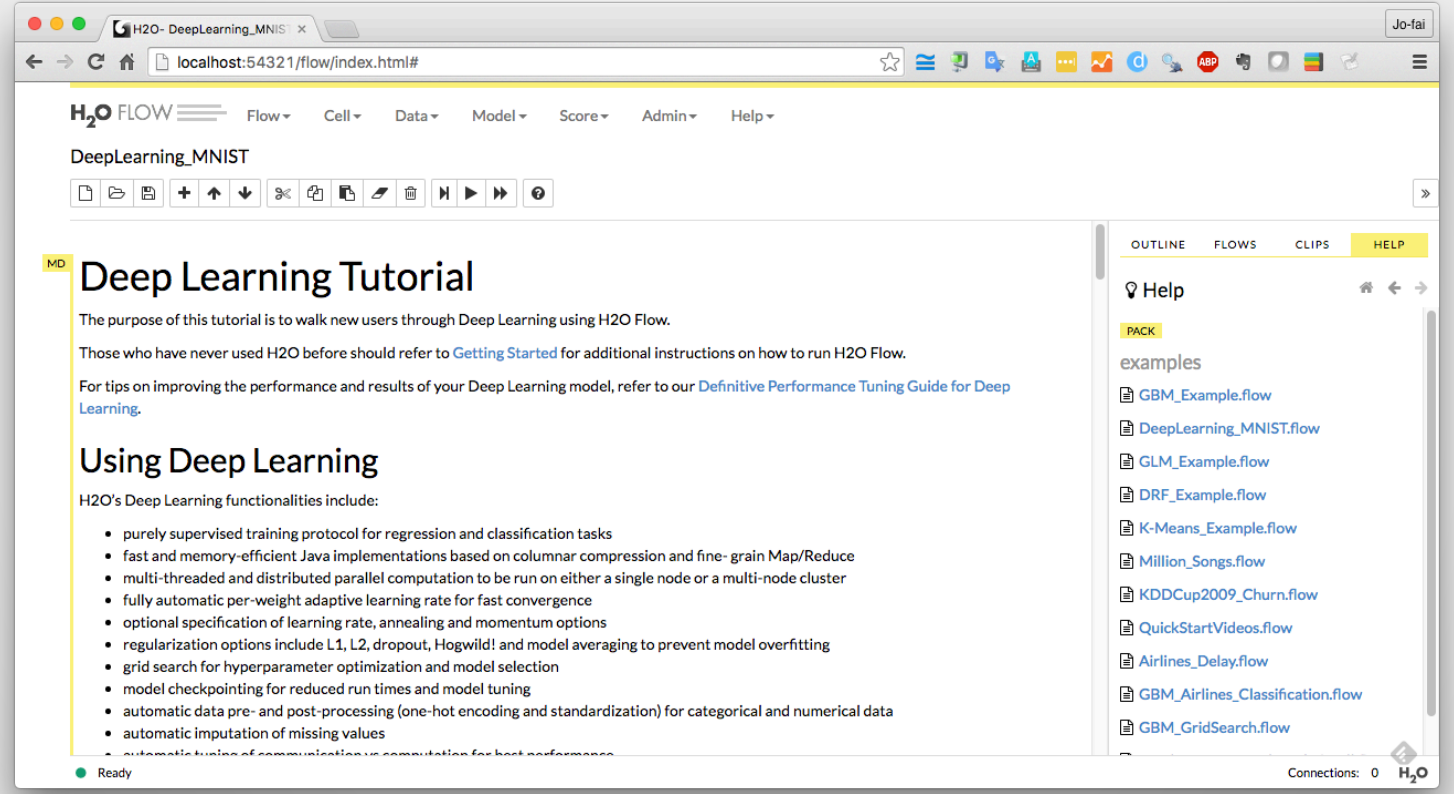
All computations are performed in highly optimized Java code in the H2O cluster, initiated by REST calls from R.

“FLOW” – H2O WEB INTERFACE

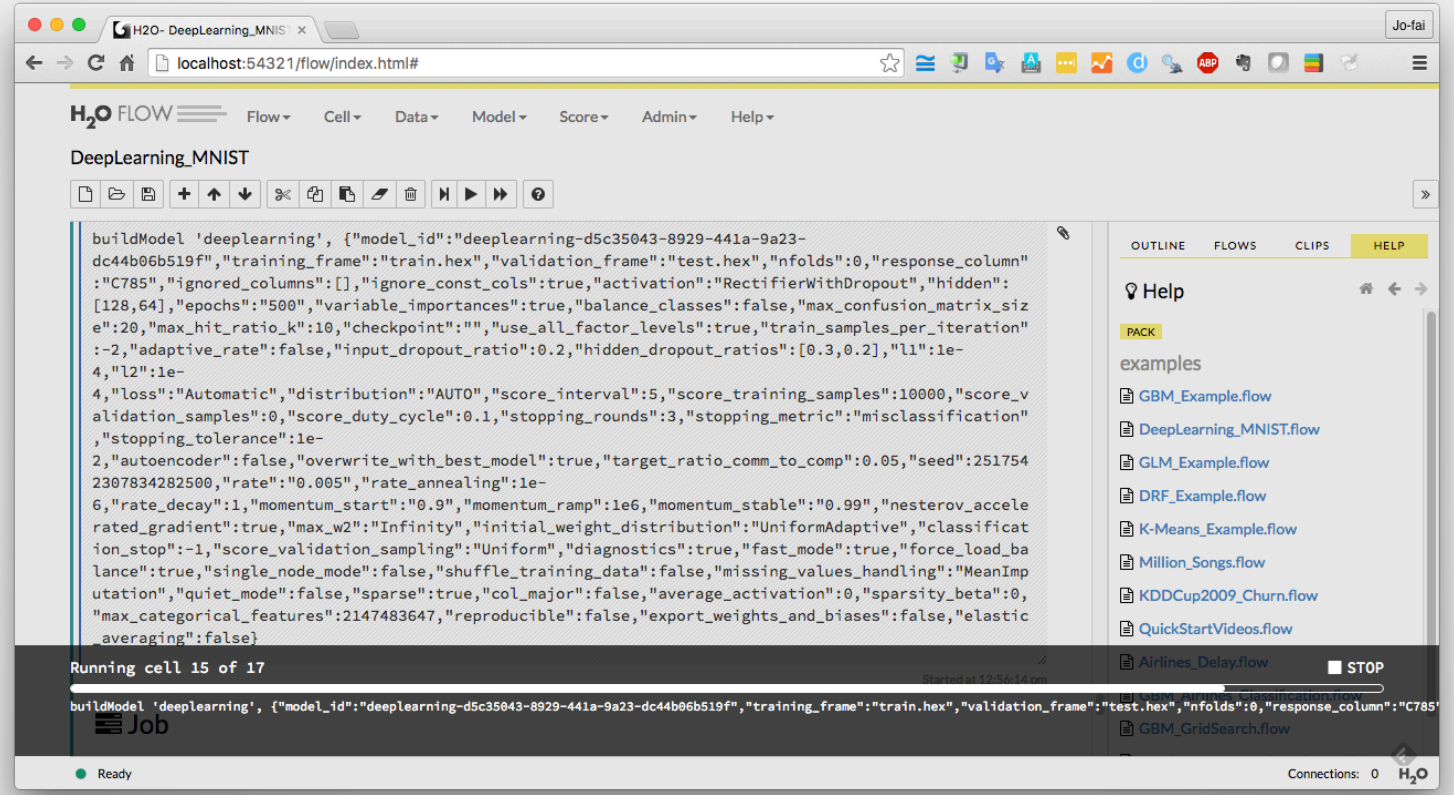
Demo – Web Interface



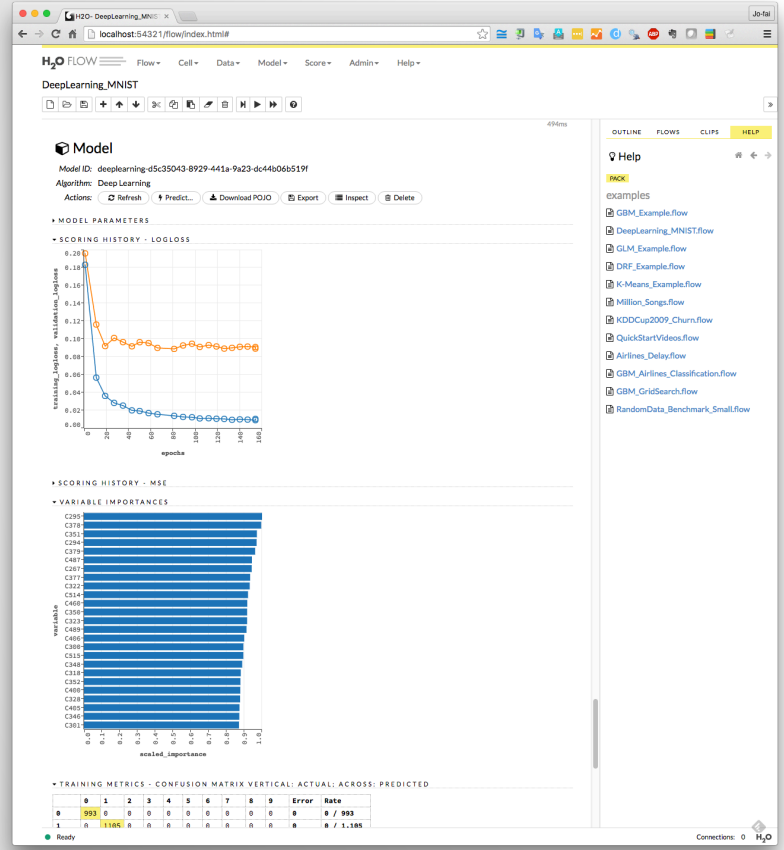
Deep Learning Tutorial

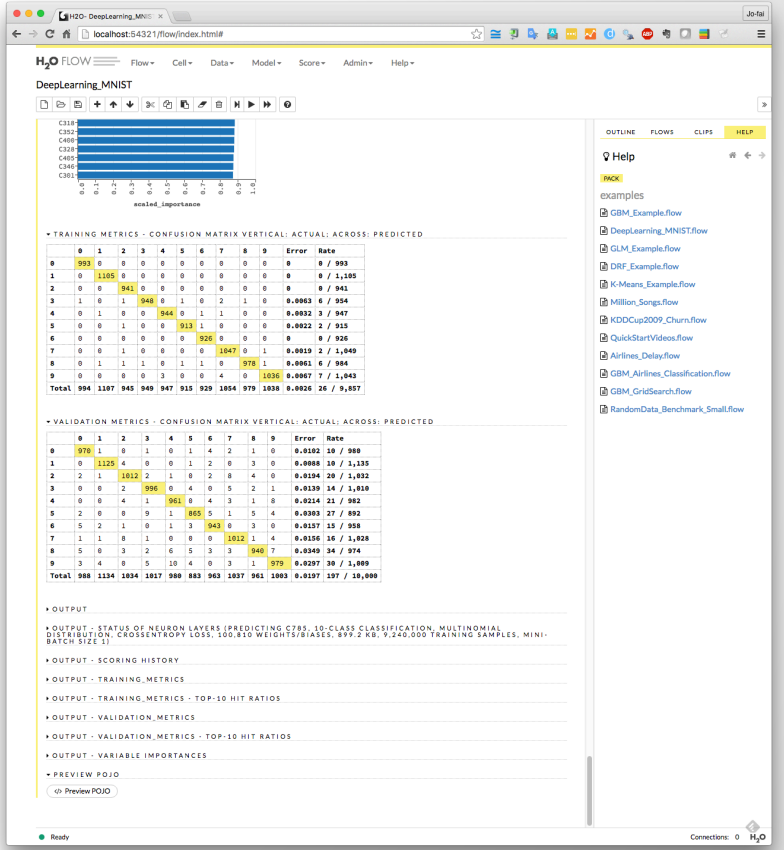


Training a Deep Learning Model

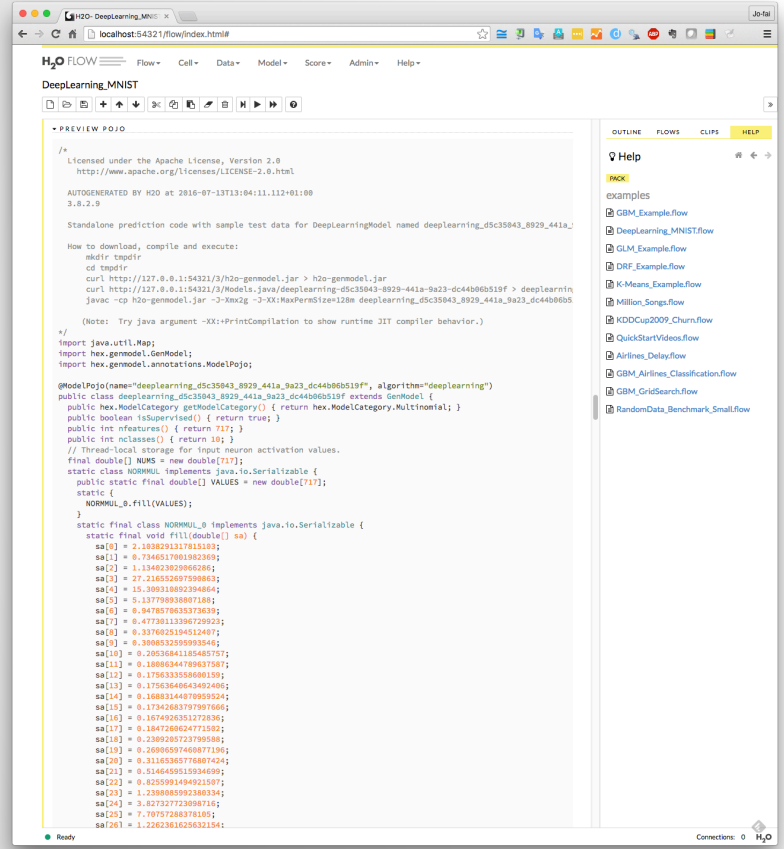


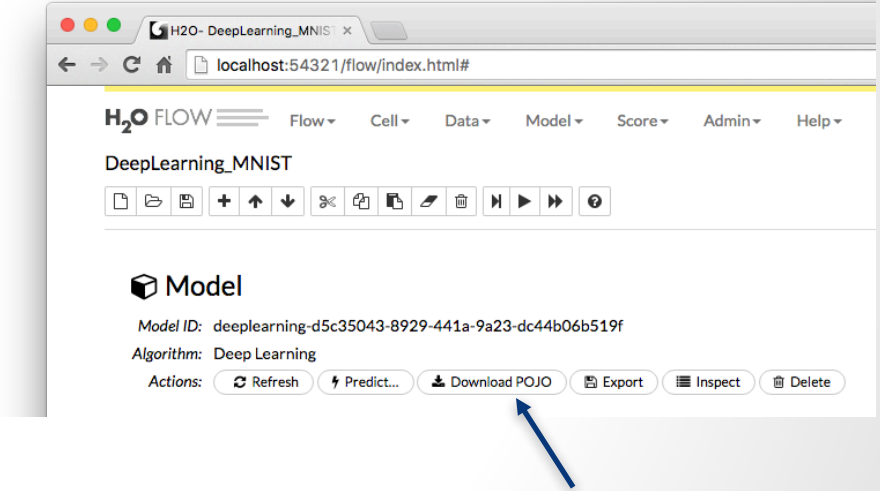
Model Evaluation





Export Plain Old Java Object (POJO)





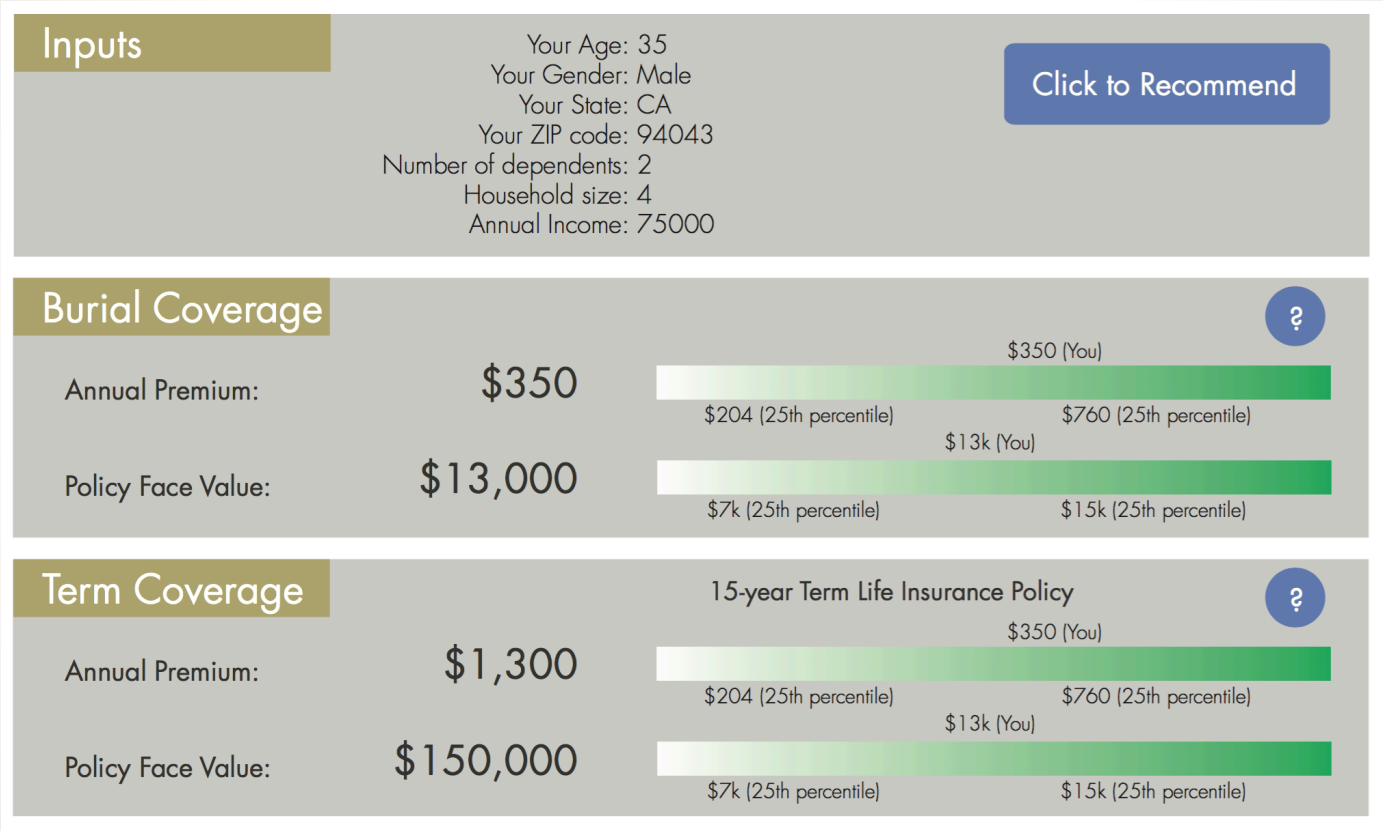
USE CASES

Use Case – Recommender System

H2O Powers Intelligent Product Recommendation Engine at Transamerica

Case Study

Use Case – Recommender System



More Use Cases

PayPal

Solving Customer Churn with Machine Learning

Case Study

H_2O.ai

KAISER PERMANENTE

Machine Learning to Save Lives

Case Study

H_2O.ai

MarketShare
a Neustar*Solution

MarketShare Enhances Its Marketing Analytics Services Using H2O

Case Study

H_2O.ai

Use Case – How People Use H2O

Arno Candel
@ArnoCandel

Following

Great use of H2O #DeepLearning for 3rd place at #Kaggle Homesite challenge! github.com/mpearmain/home… (see R scripts)

```
dl.model <- h2o.deeplearning(
  # data specifications
  x = xrange, y = max(xrange)+1, training_frame = x0.hex,
  autoencoder = FALSE,
  # network structure: activation and geometry
  activation = "RectifierWithDropout",
  hidden = c(size1, size2), epochs = 25,
  input_dropout_ratio = 0.05, hidden_dropout_ratios = c(0.05, 0.02),
  # parameters of the optimization process
  rho = 0.99, epsilon = 1e-08, rate = 0.005,
  rate_annealing = 1e-06, rate_decay = rate_dec, momentum_start = 0.5,
  l1 = 0, l2 = 0,  loss = c("CrossEntropy")
)
```

RETWEETS 18 LIKES 33

1:39 AM - 30 Jun 2016

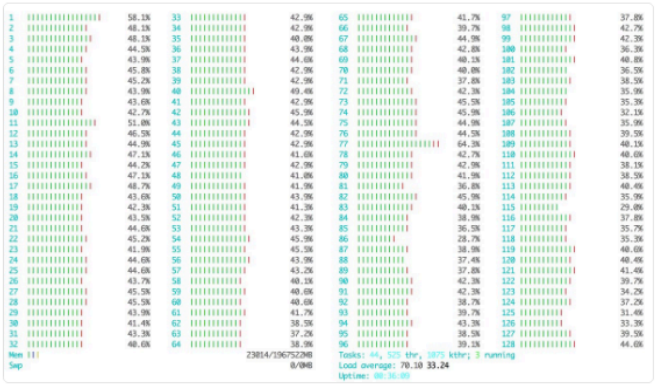
Mountain View, CA

18 33

Szilard
@DataScienceLA

Following

Also @h2oai on monster 2TB RAM 128 cores EC2 X1 #bigdata #machinelearning #datascience twitter.com/DataScienceLA/ …



RETWEETS 10 LIKES 28

5:55 PM - 24 Jun 2016

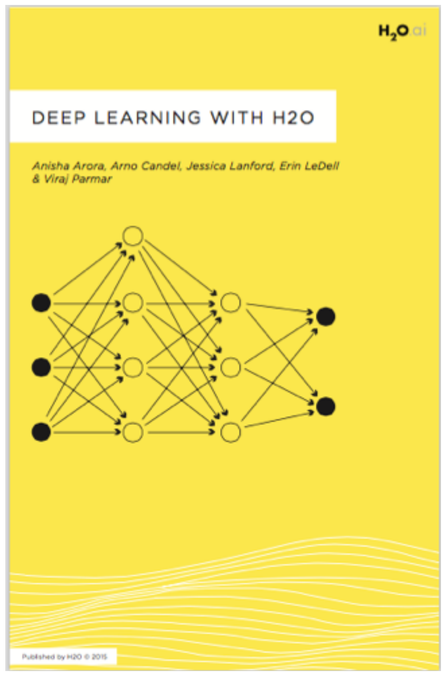
10 28

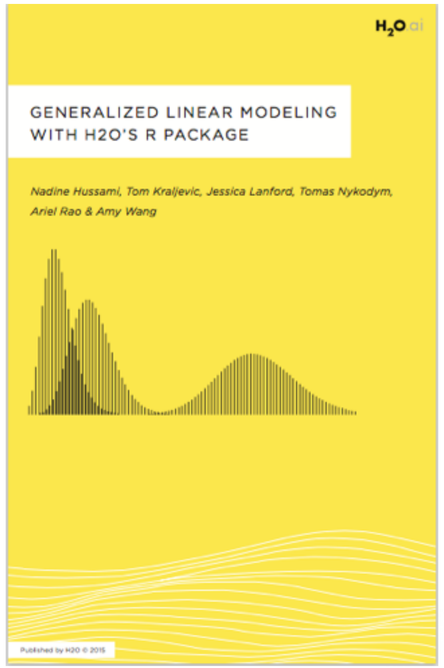
RESOURCES

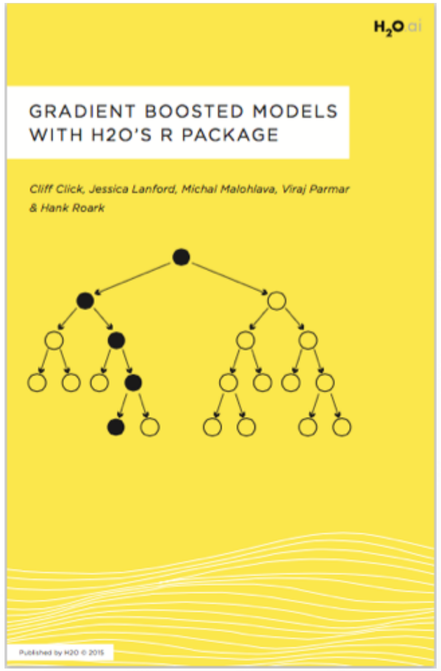
Where to learn more?

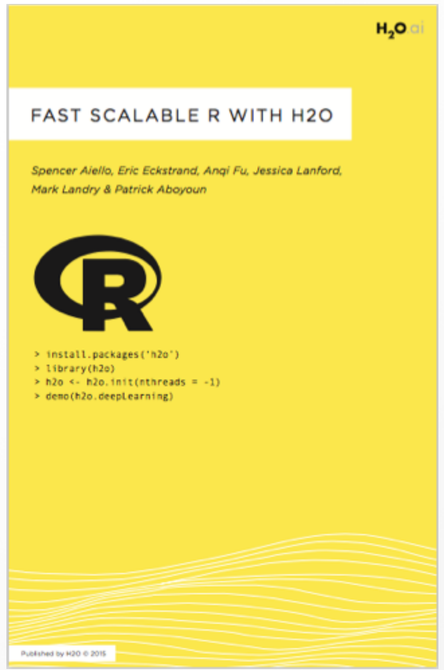
- H2O Online Training (free): http://learn.h2o.ai
- H2O Slidedecks: http://www.slideshare.net/0xdata
- H2O Video Presentations: https://www.youtube.com/user/0xdata
- H2O Community Events & Meetups: http://h2o.ai/events
- Machine Learning & Data Science courses: http://coursebuffet.com

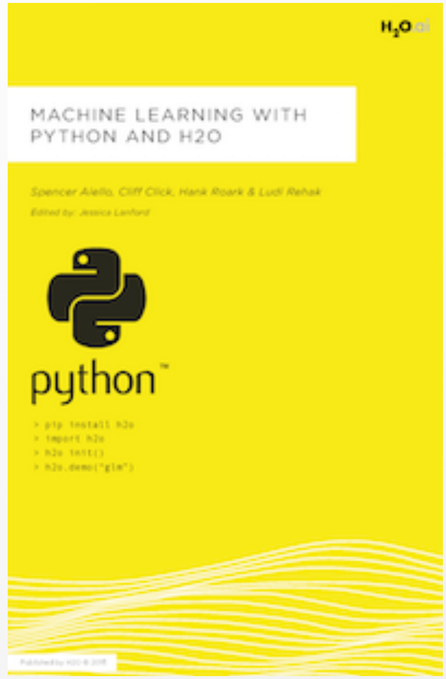
H2O Booklets











http://tinyurl.com/h2o-github-booklets

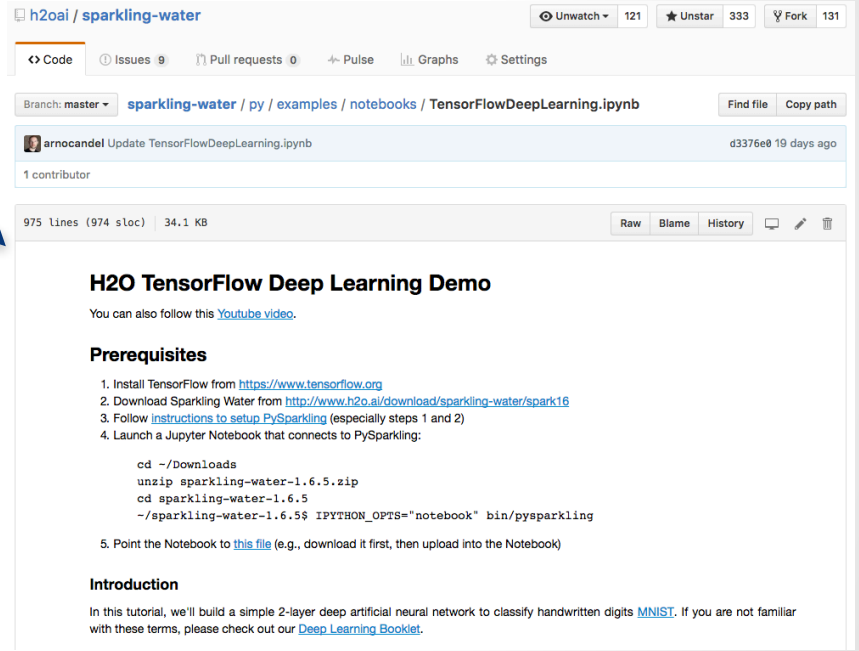
My Previous H2O Talks

- bit.ly/joe_h2o_talk1
- bit.ly/joe_h2o_talk2
- bit.ly/joe_h2o_talk3
- bit.ly/joe_h2o_talk4
- …

WHAT’S NEXT

What's Next?

- H2O + TensorFlow
- H2O + mxnet
- H2O + GPU
- H2O + x … y … z
- …



h2oai / sparkling-water

Unwatch 121 | Unstar 333 | Fork 131

Code | Issues 9 | Pull requests 0 | Pulse | Graphs | Settings

Branch: master ▾ sparkling-water / py / examples / notebooks / TensorFlowDeepLearning.ipynb — Find file | Copy path

arnocandel Update TensorFlowDeepLearning.ipynb — d3376e0 19 days ago

1 contributor

975 lines (974 sloc) | 34.1 KB — Raw | Blame | History

H2O TensorFlow Deep Learning Demo

You can also follow this Youtube video.

Prerequisites

1. Install TensorFlow from https://www.tensorflow.org
2. Download Sparkling Water from http://www.h2o.ai/download/sparkling-water/spark16
3. Follow instructions to setup PySparkling (especially steps 1 and 2)
4. Launch a Jupyter Notebook that connects to PySparkling:

```
cd ~/Downloads
unzip sparkling-water-1.6.5.zip
cd sparkling-water-1.6.5
~/sparkling-water-1.6.5$ IPYTHON_OPTS="notebook" bin/pysparkling
```

5. Point the Notebook to this file (e.g., download it first, then upload into the Notebook)

Introduction

In this tutorial, we'll build a simple 2-layer deep artificial neural network to classify handwritten digits MNIST. If you are not familiar with these terms, please check out our Deep Learning Booklet.

Any Questions?

- Contact
  - joe@h2o.ai
  - @matlabulous
  - github.com/woobe
- Links (All Slides)
  - github.com/h2oai/h2o-meetups
- H2O in London
  - Coming soon!
    - Meetups
    - Office
  - We're hiring!
  - www.h2o.ai/careers

EXTRA SLIDES

R Example – Random Grid Search

Define search range and criteria

```r
# Define a list of hyper-parameters
h_param_gbm <- list(
  max_depth       = seq(5, 10, 1),
  sample_rate     = seq(0.5, 0.8, 0.1),
  col_sample_rate = seq(0.5, 0.8, 0.1)
  )

# Search Criteria
search_criteria = list(
  strategy         = "RandomDiscrete", # Use random grid search
  max_runtime_secs = 600               # limit the runtime to 10 minutes
)
```

```r
# Grid Search
grid <- h2o.grid(
  hyper_params     = h_param_gbm,
  search_criteria = search_criteria,
  algorithm        = "gbm",           # which H2O algorithm to run

  # Standard model parameters
  training_frame   = hex_train,
  validation_frame = hex_valid,
  x                = features,
  y                = "OutcomeType",
```

```
H2O Grid Details
================

Grid ID: gbm_grid
Used hyper parameters:
  -  sample_rate
  -  max_depth
  -  col_sample_rate
Number of models: 21
Number of failed models: 0
```

Best models

```
Hyper-Parameter Search Summary: ordered by increasing logloss
  sample_rate max_depth col_sample_rate         model_ids           logloss
1         0.8         7             0.6  gbm_grid_model_2 0.752684243384921
2         0.8         6             0.5 gbm_grid_model_10  0.75318954895977
3         0.8         8             0.6  gbm_grid_model_4 0.753499220269728
4         0.7         6             0.6  gbm_grid_model_8 0.755403405990038
5         0.7         5             0.5 gbm_grid_model_14 0.756978014232093
```

```
---
   sample_rate max_depth col_sample_rate         model_ids           logloss
16         0.8        10             0.8 gbm_grid_model_12 0.765548067940223
17         0.6         9             0.7  gbm_grid_model_5 0.765760036358475
18         0.6        10             0.7 gbm_grid_model_16 0.769422559292015
19         0.5        10             0.8  gbm_grid_model_0 0.770202667369843
20         0.7        10             0.8 gbm_grid_model_17 0.773242423940347
21         0.6        10             0.5 gbm_grid_model_18 0.776840032616653
```

R Example – Model Stacking

Using h2o.stack(…) to combine multiple models

```
# H2O GBM
model_gbm ← h2o.gbm(training_frame = hex_train,
                    x = features, y = target,
                    nfolds = 5,
                    fold_assignment = "Modulo",
                    keep_cross_validation_predictions = TRUE)

# H2O Random Forest
model_drf ← h2o.randomForest(training_frame = hex_train,
                             x = features, y = target,
                             nfolds = 5,
                             fold_assignment = "Modulo",
                             keep_cross_validation_predictions = TRUE)

# Define list of models and metalearner
models ← list(model_gbm, model_drf)
metalearner ← "h2o.glm.wrapper"

# Run h2o.stack for metalearning
stack ← h2o.stack(models = models,
                  response_frame = hex_train[, target],
                  metalearner = metalearner,
                  keep_levelone_data = TRUE)
```

Getting reasonable results

Knowledge • 717 teams

Shelter Animal Outcomes

Mon 21 Mar 2016 — Sun 31 Jul 2016 (60 days to go)

Dashboard | Public Leaderboard - Shelter Animal Outcomes

This leaderboard is calculated on all of the test data.

See someone using multiple accounts? Let us know.

#	Δ1w	Team Name * In the money	Score	Entries	Last Submission UTC (Best - Last Submission)
1	–	Wag-gle *	0.45175	34	Mon, 25 Apr 2016 03:51:58 (-2.3d)
2	–	Peatle (exploits leak)	0.61647	22	Mon, 09 May 2016 09:38:23
3	–	Pau Folque	0.64808	49	Thu, 19 May 2016 13:43:08
4	–	Ryan	0.66167	2	Tue, 29 Mar 2016 03:45:47
5	–	Alexandru Papiu	0.66912	31	Fri, 20 May 2016 01:23:16
6	–	Gerard Toonstra	0.67433	8	Sat, 30 Apr 2016 21:40:31
7	–	AG12	0.68997	23	Tue, 19 Apr 2016 00:01:32 (-7.1h)
8	–	Robbin	0.69010	18	Sun, 22 May 2016 15:49:26
9	–	HaoleiFang	0.69540	35	Thu, 07 Apr 2016 23:59:39 (-0h)
10	–	Terje	0.69562	13	Thu, 26 May 2016 07:27:47 (-43d)
11	↑32	enpi	0.69785	15	Mon, 30 May 2016 17:22:31 (-46.5h)
12	↑45	Justfor	0.70015	22	Sun, 29 May 2016 23:13:28 (-24.3h)
13	↑129	MII Miguelfervi jCristobal	0.70116	29	Mon, 30 May 2016 14:58:36
14	↓3	Jason Crease	0.70313	12	Mon, 18 Apr 2016 17:10:24 (-3.1d)
15	↓3	RyosukeFukatani	0.70321	6	Mon, 23 May 2016 15:12:39
16	↓3	drewba	0.70349	75	Thu, 14 Apr 2016 16:23:46
17	↑11	**Jo-fai Chow**	**0.70417**	**7**	Thu, 26 May 2016 07:44:45
18	↓3	Brutal Ensemble	0.70434	50	Sun, 29 May 2016 15:49:14
19	↓5	Mad scientist	0.70624	23	Sat, 23 Apr 2016 09:26:36 (-7.9d)
20	↓4	jose_prueba	0.70872	5	Mon, 11 Apr 2016 17:12:37

17 out of 717 teams (≈ top 2%)