# RSparkling: Integrate H2O's Sparkling Water to your R Workflow

## Bay Area useR Group
## May 2017


Navdeep Gill, M.S.

H2O.ai


- What/who is H2O?
- H2O Platform
- H2O Sparkling Water
- Sparklyr
- RSparkling
- Demo

# H2O.ai

## H2O Company

- Team: 50. Founded in 2012, Mountain View, CA
- Stanford Math & Systems Engineers

---

## H2O Software

- Open Source Software
- Ease of Use via Web Interface
- R, Python, Scala, Spark & Hadoop Interfaces
- Distributed Algorithms Scale to Big Data


# Current Algorithm Overview

## Statistical Analysis

- Linear Models (GLM)
- Naïve Bayes

## Ensembles

- Random Forest
- Distributed Trees
- Gradient Boosting Machine
- R Package - Super Learner Ensembles

## Deep Neural Networks

- Multi-layer Feed-Forward Neural Network
- Auto-encoder
- Anomaly Detection
- Deep Features

## Clustering

- K-Means

## Dimension Reduction

- Principal Component Analysis
- Generalized Low Rank Models

## Solvers & Optimization

- Generalized ADMM Solver
- L-BFGS (Quasi Newton Method)
- Ordinary Least-Square Solver
- Stochastic Gradient Descent

## Data Munging

- Scalable Data Frames
- Sort, Slice, Log Transform

# H2O

GITTER JOIN CHAT →

H2O scales statistics, machine learning, and math over Big Data.

H2O uses familiar interfaces like R, Python, Scala, the Flow notebook graphical interface, Excel, & JSON so that Big Data enthusiasts & experts can explore, munge, model, and score datasets using a range of algorithms including advanced ones like Deep Learning. H2O is extensible so that developers can add data transformations and model algorithms of their choice and access them through all of those clients.

Data collection is easy. Decision making is hard. H2O makes it fast and easy to derive insights from your data through faster and better predictive modeling. H2O allows online scoring and modeling in a single platform.





https://github.com/h2oai/h2o-3

# H2O Components

**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 Key Value 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.
- Each node must be able to see the entire dataset (achieved using HDFS, S3, or multiple copies of the data if it is a CSV file).

# Data in H2O

**Highly Compressed**

- We read data fully parallelized from: HDFS, NFS, Amazon S3, URLs, URIs, CSV, SVMLight.
- Data is highly compressed (about 2-4 times smaller than gzip).

**Speed**

- Memory bound, not CPU bound.
- If data accessed linearly, as fast as C or Fortran.
- Speed = data volume / memory bandwidth
- ~50GB / sec (varies by hardware).

**Data Shape**

- Table width: <1k fast, <10k works, <100k slow
- Table length: Limited only by memory
- We have tested 10's of billions of rows (TBs)

# Distributed H2O Frame

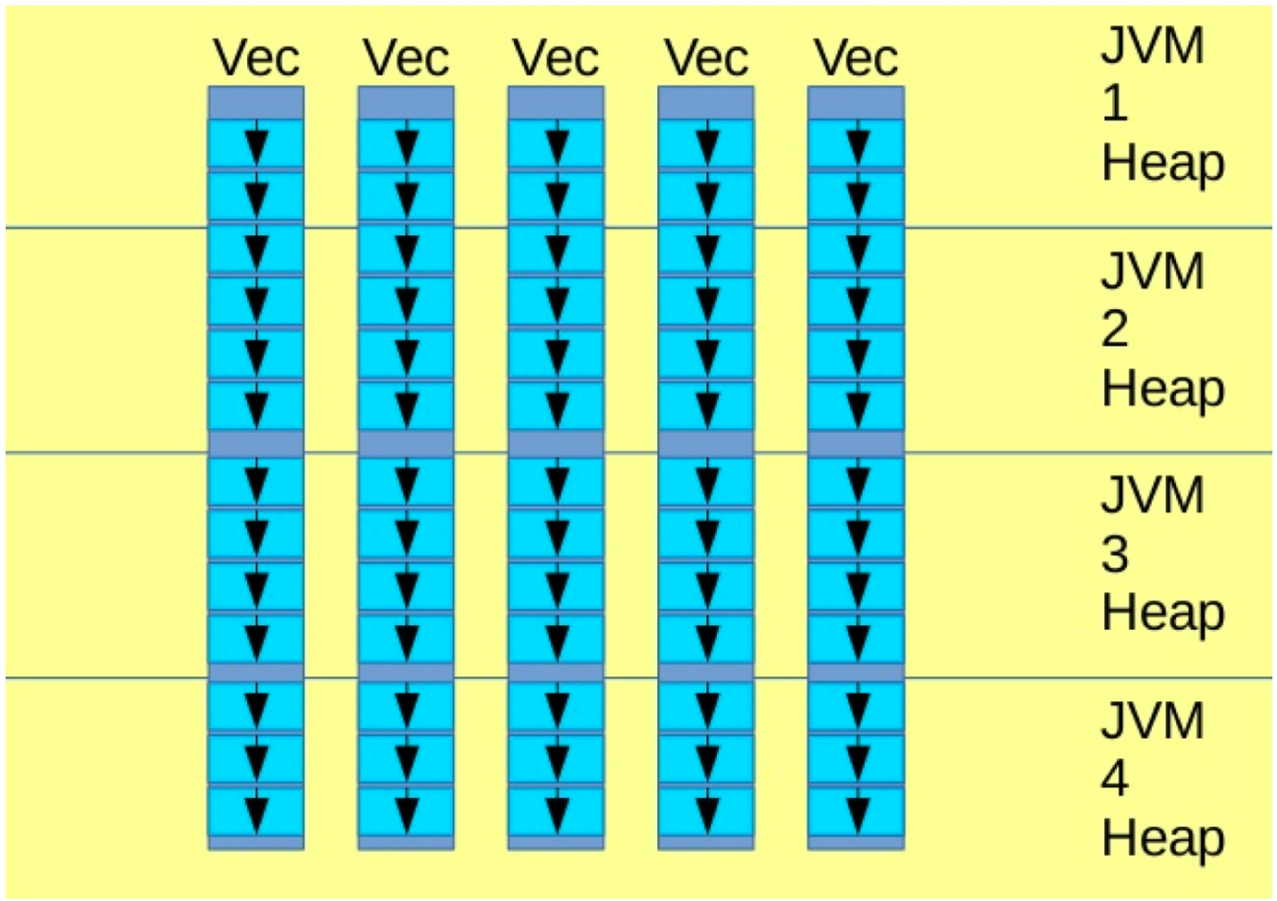



Diagram of distributed arrays. An “H2O Frame” is a collection of distributed arrays.

# Data Processing in H2O

Map Reduce

- Map/Reduce is a nice way to write blatantly parallel code (although not the only way), and we support a particularly fast and efficient flavor.
- Distributed fork/join and parallel map: within each node, classic fork / join

Group By

- We have a GroupBy operator running at scale (called ddply in the R community).
- GroupBy can handle millions of groups on billions of rows, and runs Map/Reduce tasks on the group members.

Ease of Use

- H2O has overloaded all the basic data frame manipulation functions in R and Python.
- Tasks such as imputation and one-hot encoding of categoricals is performed inside the algorithms.

# H2O in Spark

Spark + $H_2O$

SPARKLING WATER

# H2O Sparkling Water

## Spark Integration

- Sparkling Water is transparent integration of H2O into the Spark ecosystem.
- H2O runs inside the Spark Executor JVM.

## Benefits

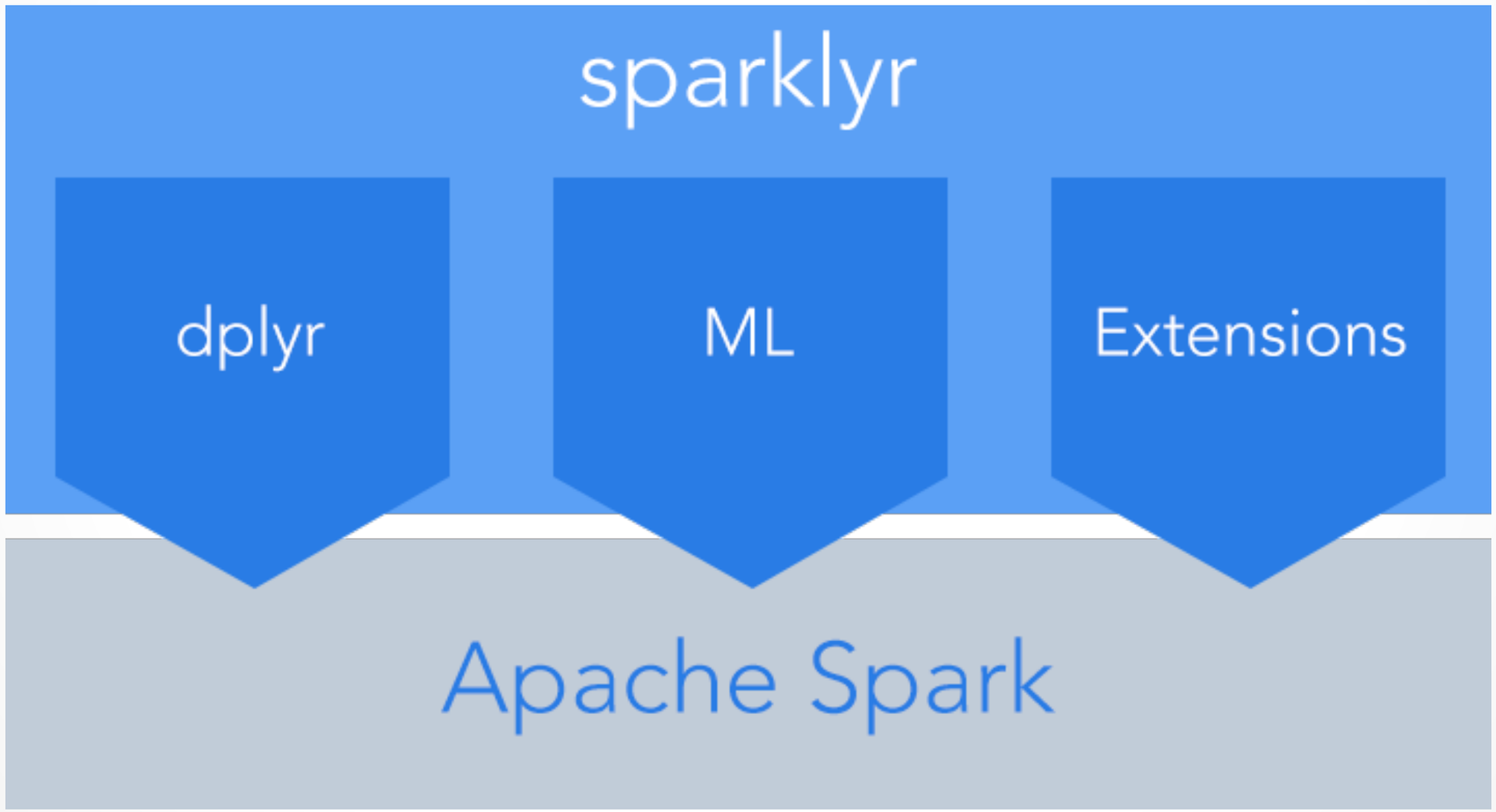
- Provides advanced machine learning algorithms to Spark workflows.
- Sophisticated alternative to the default MLlib library in Spark.

## Sparkling Shell

- Sparkling Shell is just a standard Spark shell with additional Sparkling Water classes
- `export MASTER="local-cluster[3,2,1024]"`
- `spark-shell –jars sparkling-water.jar`

# Sparklyr

**“SPARRK-lee-ARR”**



# Sparklyr

- Connect to Spark from R.
- The sparklyr package provides a complete dplyr backend.
- Filter and aggregate Spark datasets then bring them into R for analysis and visualization.
- Use Spark's distributed machine learning library from R.
- Create extensions that call the full Spark API and provide interfaces to Spark packages.

```
library(sparklyr)
spark_install()
sc <- spark_connect("local")
my_tbl <- copy_to(sc,iris)
```

# RSparkling

+

Spark + $H_2O$

+

SPARKLING WATER

# RSparkling

- The rsparkling R package is an extension package for sparkapi / sparklyr that creates an R front-end for a Spark package (Sparkling Water from H2O) .
- This provides an interface to H2O's machine learning algorithms on Spark, using R.
- This package implements basic functionality (creating an H2OContext, showing the H2O Flow interface, and converting between Spark DataFrames and H2O Frames).

```
library(sparklyr)
spark_install(version = "2.0.0")
options(rsparkling.sparklingwater.version =
"2.0.0")
library(rsparkling)
sc <- spark_connect(master = "local")
```

# DEMO!

# Thank you!

@Navdeep_Gill_ on Twitter
navdeep-G on GitHub
navdeep@h2o.ai
Demo: https://github.com/h2oai/sparkling-water/blob/master/r/examples/nycflights13.R
Sparkling Water: https://github.com/h2oai/sparkling-water
RSparkling: https://github.com/h2oai/sparkling-water/tree/master/r
Sparklyr: https://github.com/rstudio/sparklyr

# Appendix: New & Active Developments in H2O

# DeepWater

- Native implementation of Deep Learning models for GPU-optimized backends (mxnet, Caffe, TensorFlow, etc.)
- State-of-the-art Deep Learning models trained from the H2O Platform
- Provides an easy to use interface to any of the Deep Water backends.
- Extends the H2O platform to include Convolutional Neural Nets (CNNs) and Recurrent Neural Nets (RNNs) including LSTMs

# AutoML

- AutoML stands for "Automatic Machine Learning"
- The idea here is to remove most (or all) of the parameters from the algorithm, as well as automatically generate derived features that will aid in learning.
- Single algorithms are tuned automatically using a carefully constructed random grid search (future: Bayesian Optimization algorithms).
- Optionally, a Stacked Ensemble can be constructed.
- The current version of AutoML trains and cross validates a Random Forest, an Extremely Random Forest, a random grid of GBM's, a random grid of Deep Neural Nets, and a Stacked Ensemble of all the models.
- Beta version is available in the h2o-3 master branch.