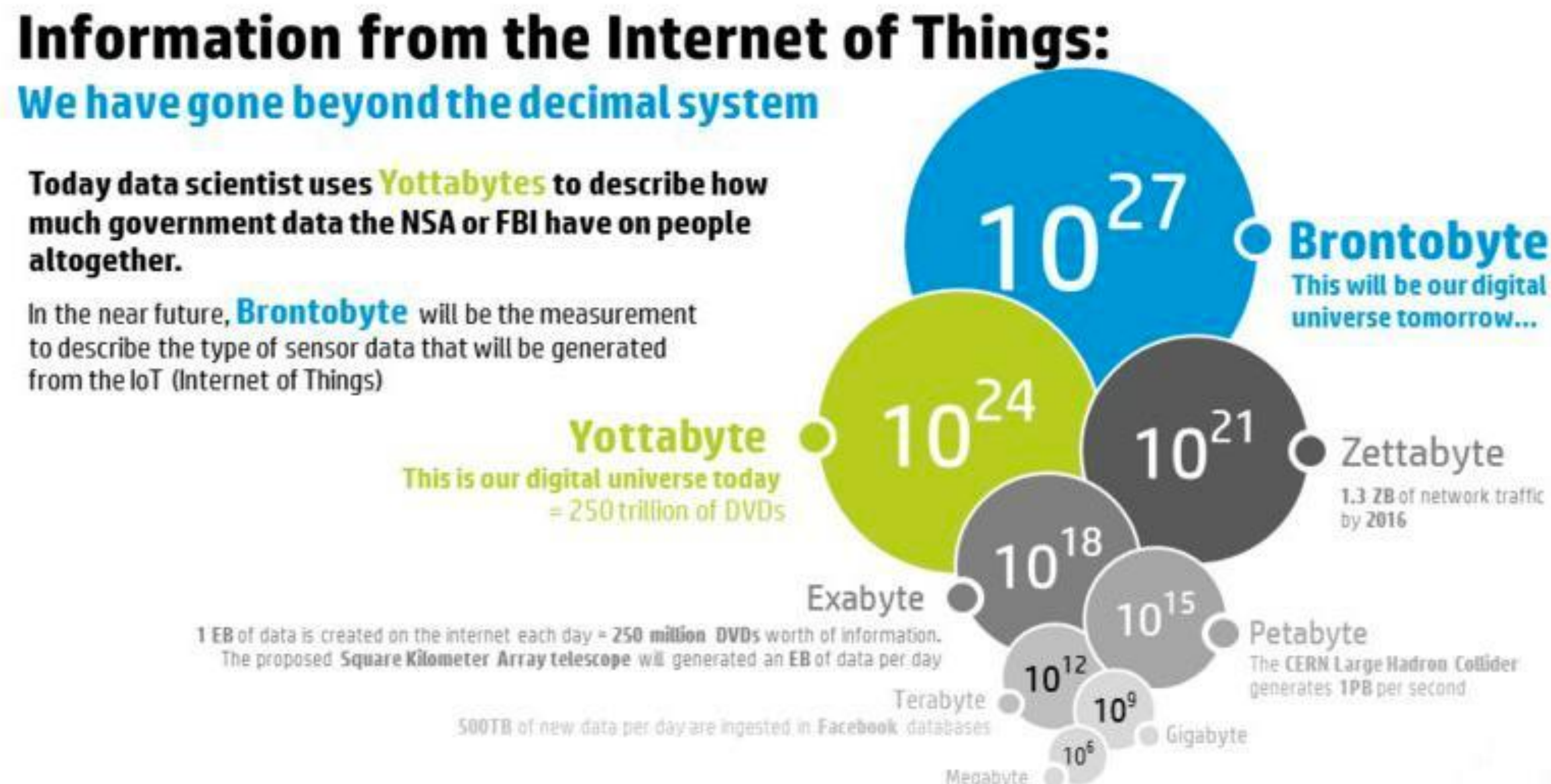
Deep Learning for Internet of Things Application Using H2O Platform

Basheer Qolomany

CS6030: Internet of Things – Application Development

Internet of Things (IoT) is heavily signal data

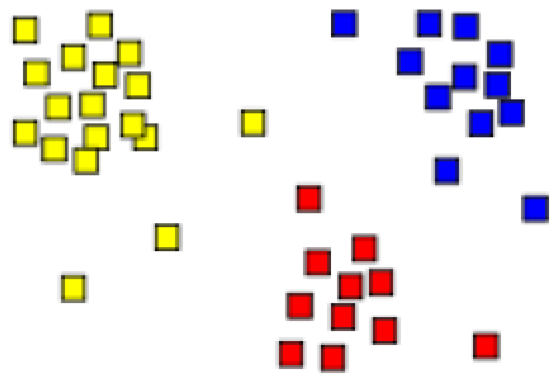


Machine Learning -Definition

- A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data.
- *“The ability of a program to learn from experience—that is, to modify its execution on the basis of newly acquired information.”*

What is Clustering?

- **Clustering:**is the assignment of a set of observations into subsets (called *clusters*) so that observations in the same cluster are similar in some sense. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields.



What is Classification?

- Classification is the task of *learning a target function* f that maps attribute set x to one of the predefined class labels y

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

One of the attributes is the class attribute
In this case: Cheat

Two class labels (or classes): Yes (1), No (0)

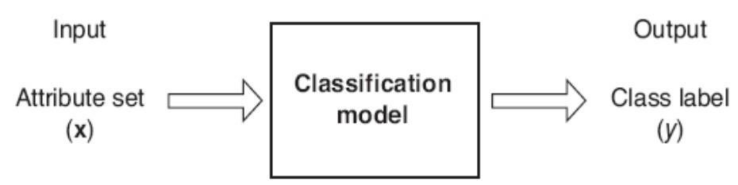


Figure 4.2. Classification as the task of mapping an input attribute set $\mathbf{x}$ into its class label y.

Catching tax-evasion

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Tax-return data for year 2011

A new tax return for 2012
Is this a cheating tax return?

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

An instance of the classification problem: learn a method for discriminating between records of different classes (cheaters vs non-cheaters)

Why Classification?

- The target function f is known as a classification model
- Descriptive modeling: Explanatory tool to distinguish between objects of different classes (e.g., understand why people cheat on their taxes)
- Predictive modeling: Predict a class of a previously unseen record

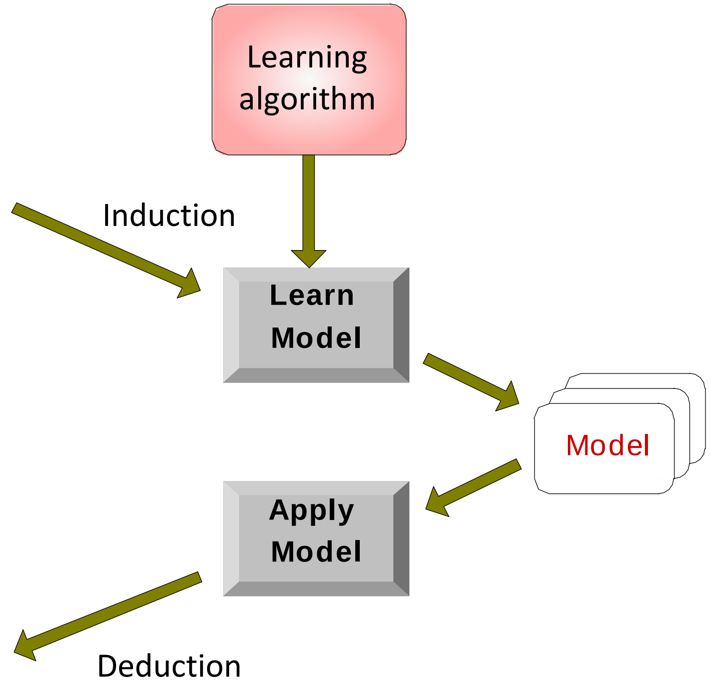
Illustrating Classification Task

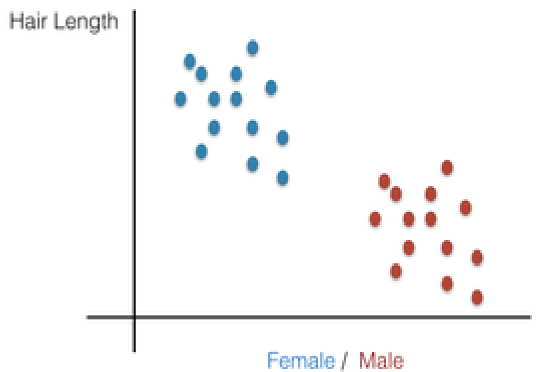
Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

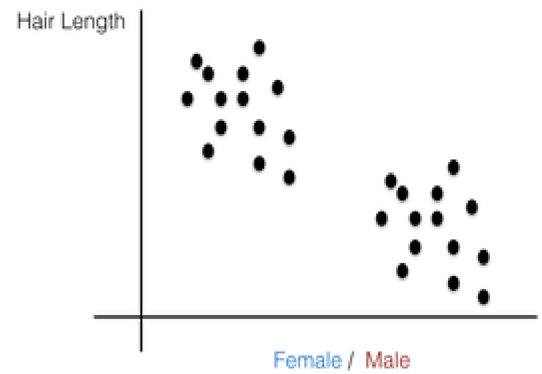
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



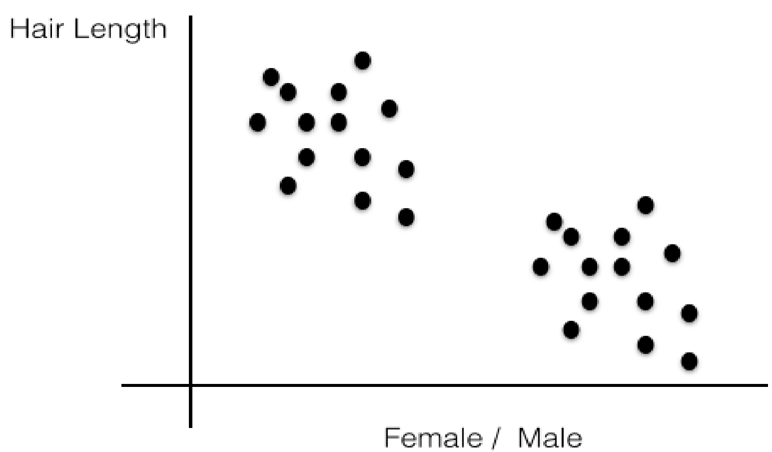


Learn



Apply

In classification, you first 'Learn' what goes with what and then you 'Apply' that knowledge to new examples. So if somebody gave us the first picture on the left, which is a plot of hair length (Y axis) against gender (on X axis)



Infer

In this case, clustering algorithm has to "Infer" that you could create at least two groups of points.

The curse of dimensionality

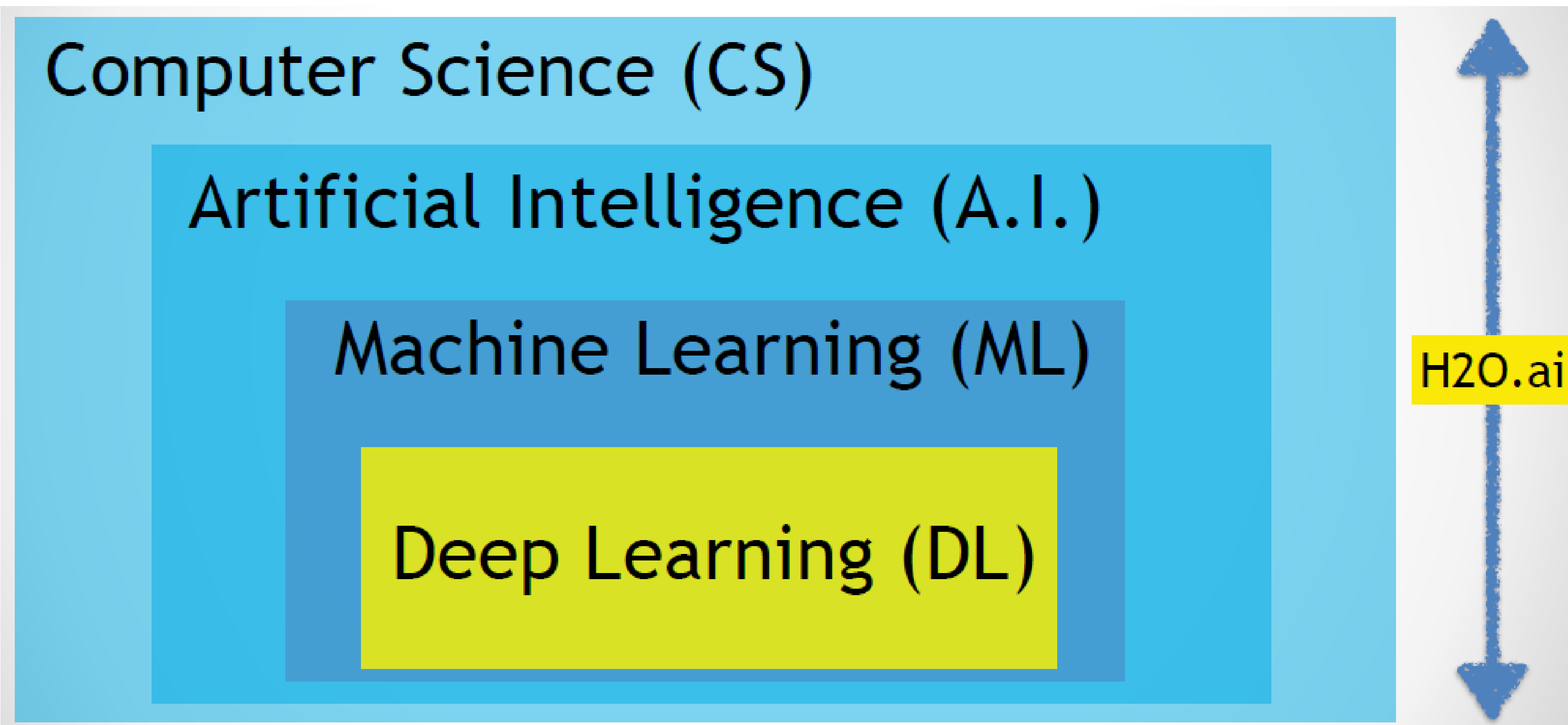
- Real data usually have thousands, or millions of dimensions
  - E.g., web documents, where the dimensionality is the vocabulary of words
  - Facebook graph, where the dimensionality is the number of users
- Huge number of dimensions causes problems
  - Data becomes very sparse, some algorithms become meaningless (e.g. density based clustering)
  - The complexity of several algorithms depends on the dimensionality and they become infeasible.

Dimensionality reduction

- In machine learning and statistics, dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration, via obtaining a set of "uncorrelated" principal variables.
- Usually the data can be described with fewer dimensions, without losing much of the meaning of the data.
- Essentially, we assume that some of the data is noise, and we can approximate the useful part with a lower dimensionality space.
  - Dimensionality reduction does not just reduce the amount of data, it often brings out the useful part of the data

What is Deep Learning?

- it's a computer algorithm that models high-level abstractions in data with multiple layers of non-linear transformations.



Gartner Identifies the Top 10 Strategic Technology Trends for 2016

http://www.gartner.com/newsroom/id/3143521 Oct 6, 2015

Gartner Tech Trends	Description	Relates to this talk
Advanced Machine Learning	Deep Neural Nets	DNN to solve application needs
Device Mesh	Mobile, wearable, home, auto, IoT	Practical applications, input data
Adaptive Security Architecture	Move from static rules and patterns to understand user and systems	Practical applications
Information of Everything	Contextual, integrated	Input data
Ambient User Experience	Over environments, time location	Output to users (i.e. real time scoring)
Autonomous Agents and Things	Smart advisors	Output to users (i.e. real time scoring)
Advanced System Architectures	Train DNN with GPUs and FPGAs, cloud architectures	Train DNN
Mesh App and Service Architecture	3 tier --> loosely coupled apps and services for web scale performance & flexibility	Application deployment architecture
Internet of Things Platforms	Complements mesh app and service arch, implmenetations of IoT	Input data system integrating with deployment architecture
3D Printing	Expect annual growth rate of 64% for enterprise printers through 2019	n/a

Gartner Identifies the Top 10 Internet of Things Technologies for 2017 and 2018

- IoT Security
- IoT Analytics
- IoT Device Management
- Low-Power, Short-Range IoT Networks
- Low-Power, Wide-Area Networks
- IoT Processors
- IoT Operating Systems
- Event Stream Processing
- IoT Platforms
- IoT Standards and Ecosystems

Source: [Gartner Identifies the Top 10 Internet of Things Technologies for 2017 and 2018]
Published February 23, 2016

What problems can deep machine learning address?

- Spam Detection
- Credit Card Fraud Detection
- Digit Recognition
- Speech Understanding
- Face Detection
- Product Recommendation
- Medical Diagnosis
- Stock Trading
- Customer Segmentation
- Shape Detection

Step 1: Great Algorithms + Fast Computers

- Raw computing power can automate complex tasks!

1997: Playing Chess
(IBM Deep Blue beats Kasparov)

Earlier he said: “No computer will ever beat me.”

Computer Science

30 custom CPUs, 60 billion moves in 3 mins

Step 2: More Data + Real-Time Processing

- Automating automobiles into autonomous automata!

2005: Self-driving Cars
DARPA Grand Challenge, 132 miles
(won by Stanford A.I. lab*)

Sensors & Computer Science
video, radar, laser, GPS, 7 Pentium computers

Step 3: Big Data + In-Memory Clusters

- Automating question answering and information retrieval!

2011: Jeopardy (IBM Watson)

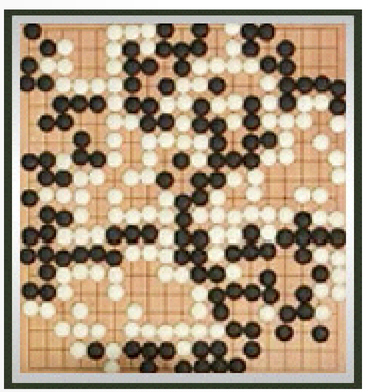
In-Memory Analytics/ML

4 TB of data (incl. wikipedia), 90 servers, 16 TB RAM, Hadoop, 6 million logic rules

- Note: IBM Watson received the question in electronic written form, and was often able to (electronically) press the answer button faster than the competing humans.

Step 4: Deep Learning

- Deep Learning + Smart Algorithms = Master Gamer.



2014: Atari Games (DeepMind)
trained from raw pixel values, no human rules

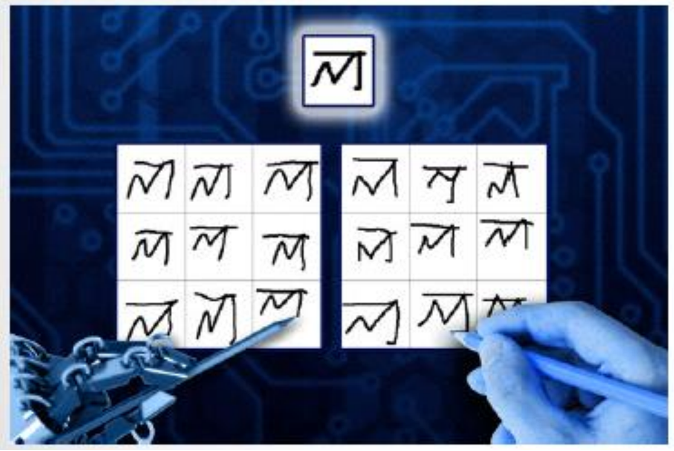
Deep Learning

+ reinforcement learning, tree search,
Monte Carlo, GPUs, playing against itself, ...

2016: AlphaGo (Google DeepMind)

Step 5: Improve Training Efficiency

- New algorithm learns handwriting of unseen symbols from very few training examples (unlike typical Deep Learning).



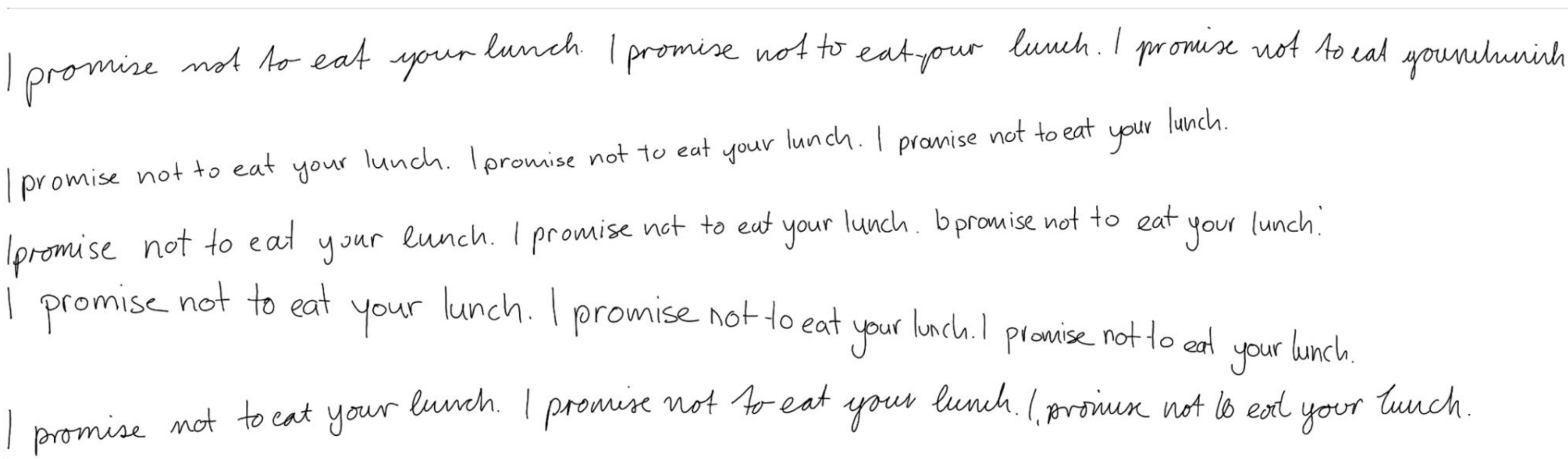
2015: MIT, NYU, Toronto

Bayesian Program Learning

(NOT Deep Learning)

What ELSE can Deep Learning do?

- Deep Learning can generate handwriting



What ELSE can Deep Learning do?

- Deep Learning can generate code, captions, language, etc.
- Generated math proof:

$$
\begin{array}{ccc}
= \alpha' \longrightarrow & & \\
\updownarrow & & \\
= \alpha' \longrightarrow \alpha & & X \\
 & & \downarrow \\
\mathrm{Spec}(K_\psi) & & \mathrm{Mor}_{Sets} \quad \mathrm{d}(\mathcal{O}_{X_{X/k}}, \mathcal{G})
\end{array}
$$

is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \ \text{-}1(\mathcal{O}_{X_{etale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.

Image captioning:

"black and white dog jumps over bar."

What ELSE can Deep Learning do?

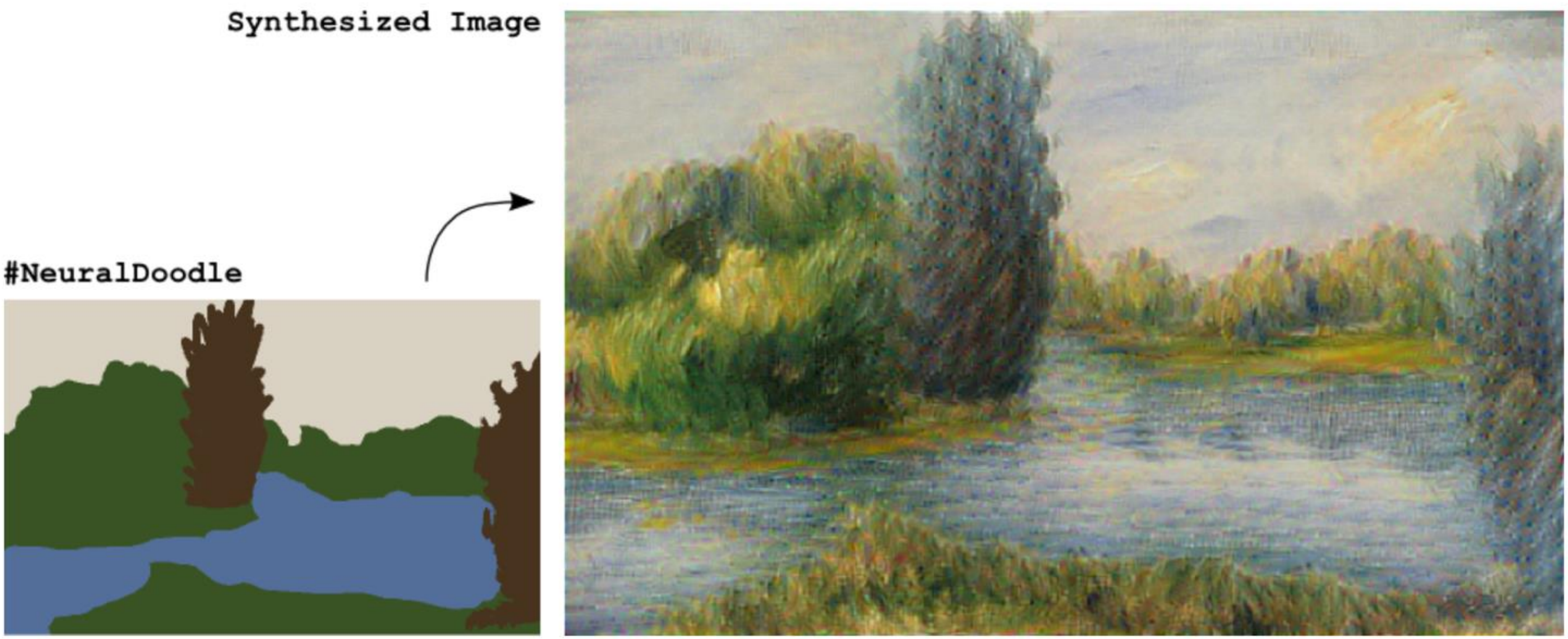
- Deep Learning can translate any language

Quest Visual (acquired by Google)

Google Translate
By Google, Inc.

What ELSE can Deep Learning do?

- Deep Learning can create masterpieces: Semantic Style Transfer



Deep Learning Tools

Trained model may end up with same accuracy. Choose by develop. ease & production

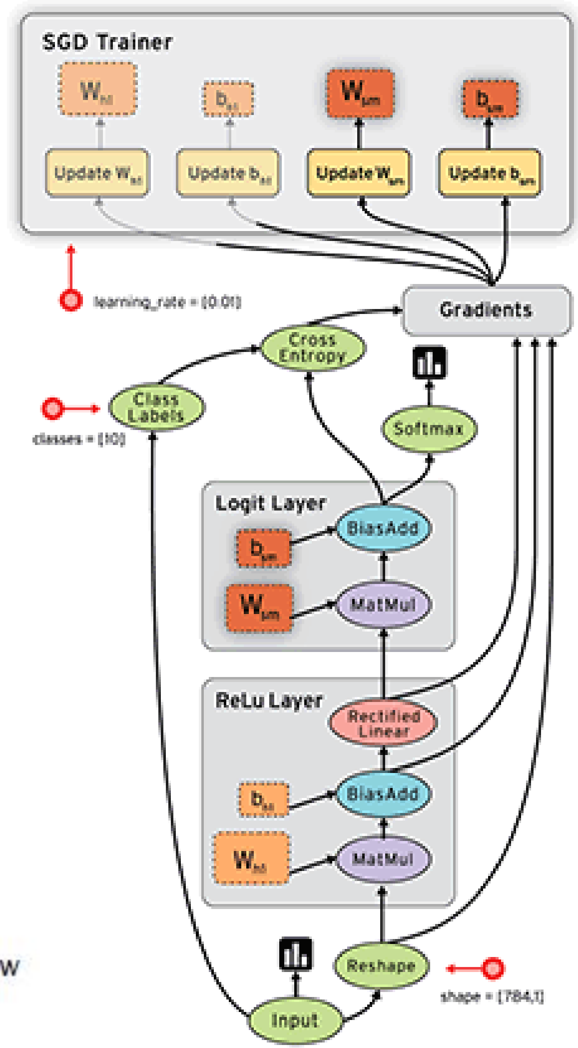
System or Package	Language	Algorithms and Comments	Org Support
Theano	Python, NumPy scripting	Python expressions in Theano, can compile to C with g++, use GPU (CUDA), parallel execution. Oldest and most used mature deep learning system.	Univ of Montreal
Torch7	Lua scripting, command line interpreter	N-dimensional array or Tensor. Used by Google DeepMind, Facebook AI, IBM, Yandex. On Android and iOS	
TensorFlow	Python scripting, C++ back	Tensors + flow graphs. 2nd generation deep learning at Google, just came out Nov 2015. CPU, GPU & mobile. Integrated network GUI. Jeff Dean	Google

Deep Learning Tools

TensorFlow

By Google, 600 DL proj
- Speech
- Google Photos
- Translation
- Gmail
- Search

Rajat Monga, Tech Lead & Manager for TensorFlow



Deep Learning Tools

System or Package	Language	Algorithms and Comments	Org Support
H2O	Java, Python, R Scala , JSON	autoencoder, convolutional, parallel weight updates over servers	H2O.ai
DL4J DeepLearning4j	Java, Scala	autoencoder, deep belief network, DBN, word vector, recurrent net	Skymind
Gorila		General Reinforcement Learning Architecture, David Silver	Google DeepMind
Kamanja	18 data mining systems generate PMML. Also Java & Scala	Various neural net training systems, such as R. Good compute framework for scaling up full system, end to end.	LigaDATA
deepnet	R	BP, RBM, DBN, autoencoder	
darch	R	Restricted Boltzman Machines (RBM)	
autoencoder	R	Sparse autoencoder, from Ng's notes - for deep belief neural nets	

What is H2O?

Math Platform — Open source in-memory prediction engine

- Parallelized and distributed algorithms making the most use out of multithreaded systems
- GLM, Random Forest, GBM, PCA, etc.

API — Easy to use and adopt

- Written in Java – perfect for Java Programmers
- REST API (JSON) – drives H2O from R, Python, Java, Scala, Excel, Tableau

Big Data — More data? Or better models? BOTH

- Use all of your data – model without down sampling
- Run a simple GLM or a more complex GBM to find the best fit for the data
- More Data + Better Models = Better Predictions

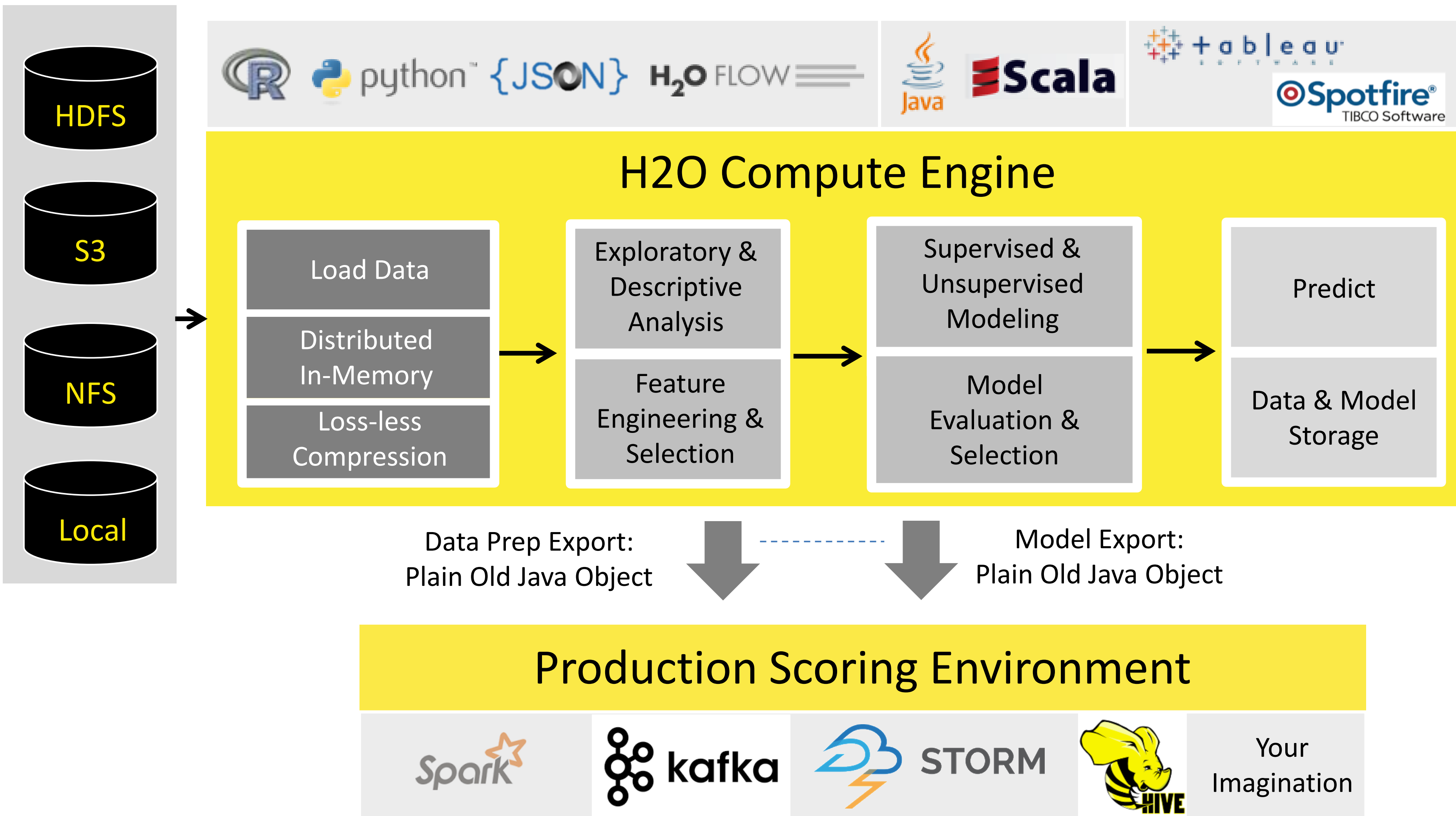
H2O Platform Overview

- 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 Production Analytics Workflow



Algorithms on H_2O

Supervised Learning

Statistical Analysis

- **Generalized Linear Models with Regularization**: 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

Algorithms on H_2O

Unsupervised Learning

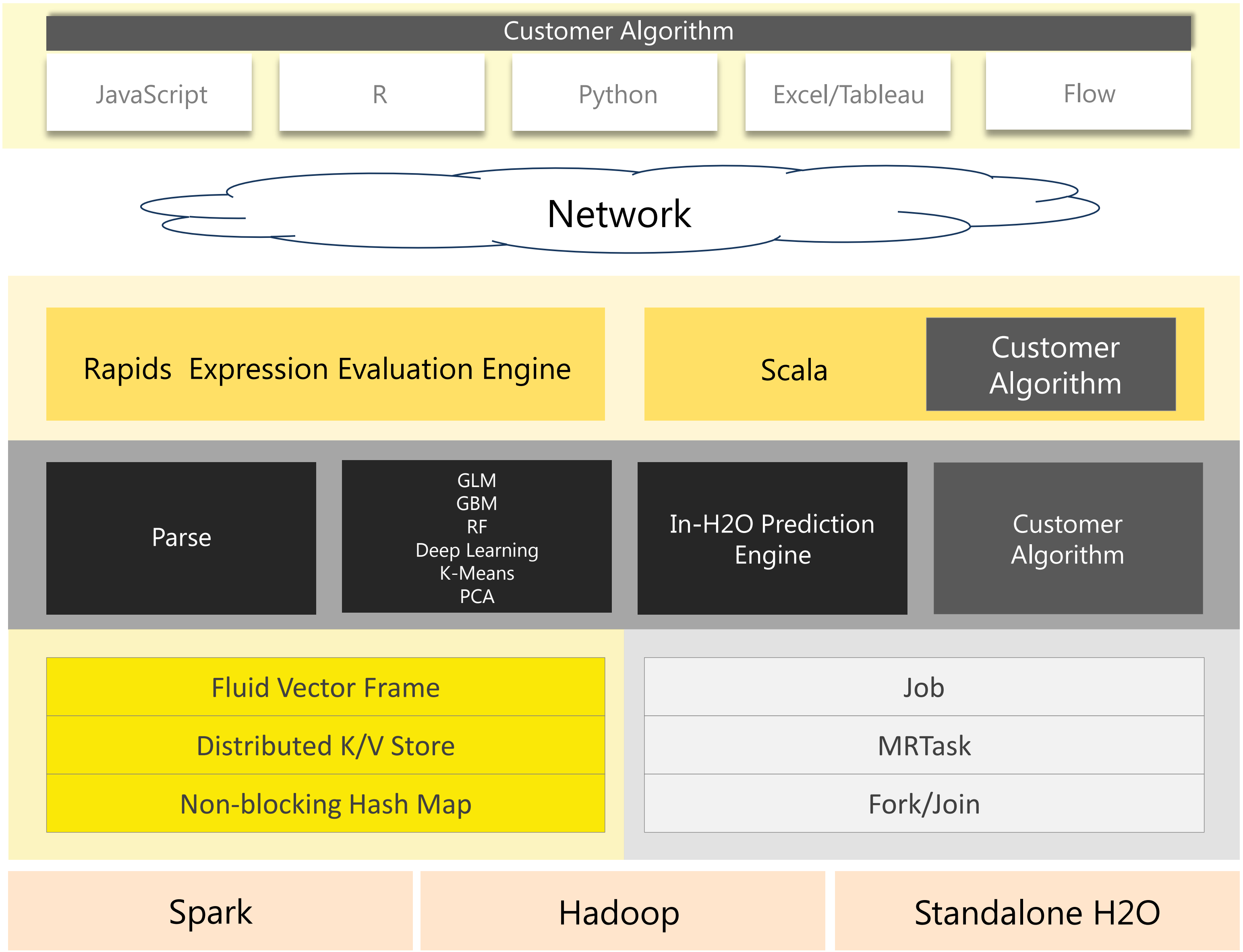
Clustering

Dimensionality Reduction

Anomaly Detection

- **K-means**: Partitions observations into k clusters/groups of the same spatial size
- **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
- **Autoencoders**: Find outliers using a nonlinear dimensionality reduction using deep learning

H2O Software Stack



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).

Distributed K/V Store

Peer-to-Peer

- The H2O K/V Store is a classic peer-to-peer distributed hash table.
- There is no “name-node” nor central key dictionary.

Pseudo-Random Hash

- Each key has a home-node, but the homes are picked pseudo-randomly per-key.
- This allows us to force keys to “home” to different nodes (usually for load-balance reasons).

Key’s Home Node

- A key's “home” is solely responsible for breaking ties in racing writes and is the “source of truth.”
- Keys can be cached anywhere, and both reads & writes can be cached (although a write is not complete until it reaches “home”.)

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

H2O and R

What is R?

- The R statistical programming language is a free open source package based on the S language developed by Bell Labs.
- The language is very powerful for writing programs.
- Many statistical functions are already built in. It includes routines for data summary and exploration, graphical presentation and data modelling.
- Contributed packages expand the functionality to cutting edge research.
- Since it is a programming language, generating computer code to complete tasks is required.

How to download?

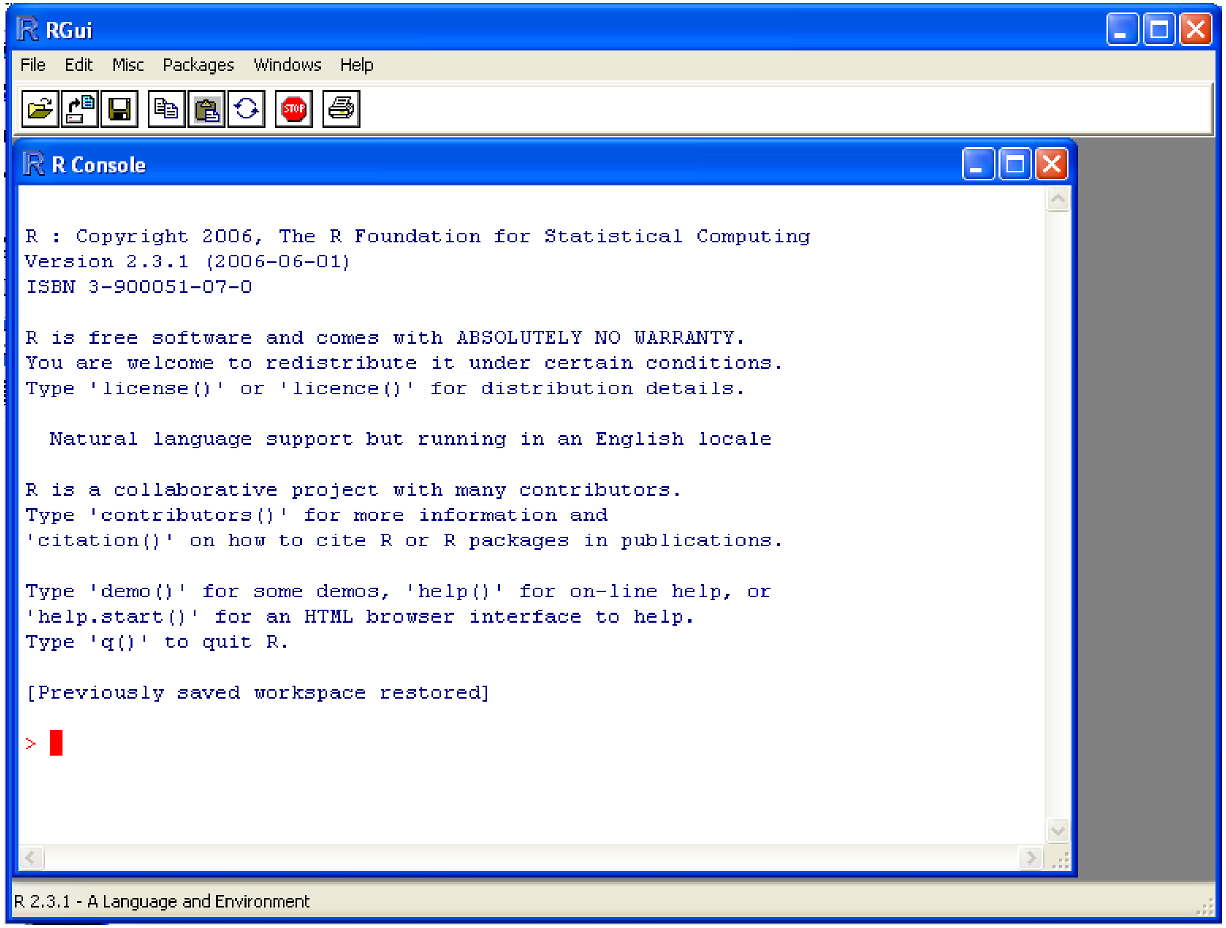
– Google it using R or CRAN

(Comprehensive R Archive Network)

– http://www.r-project.org

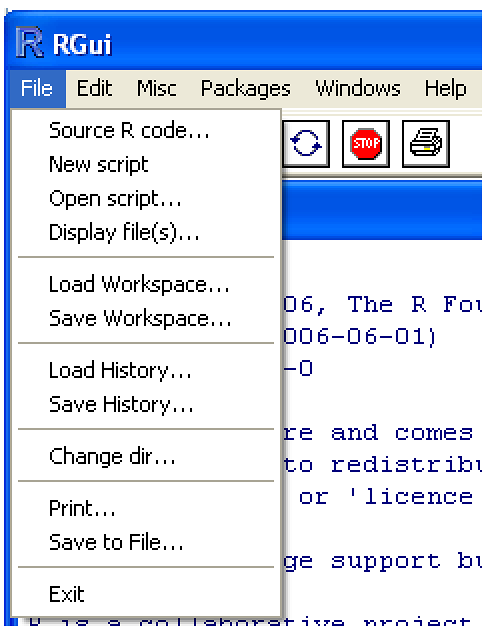
Getting Started

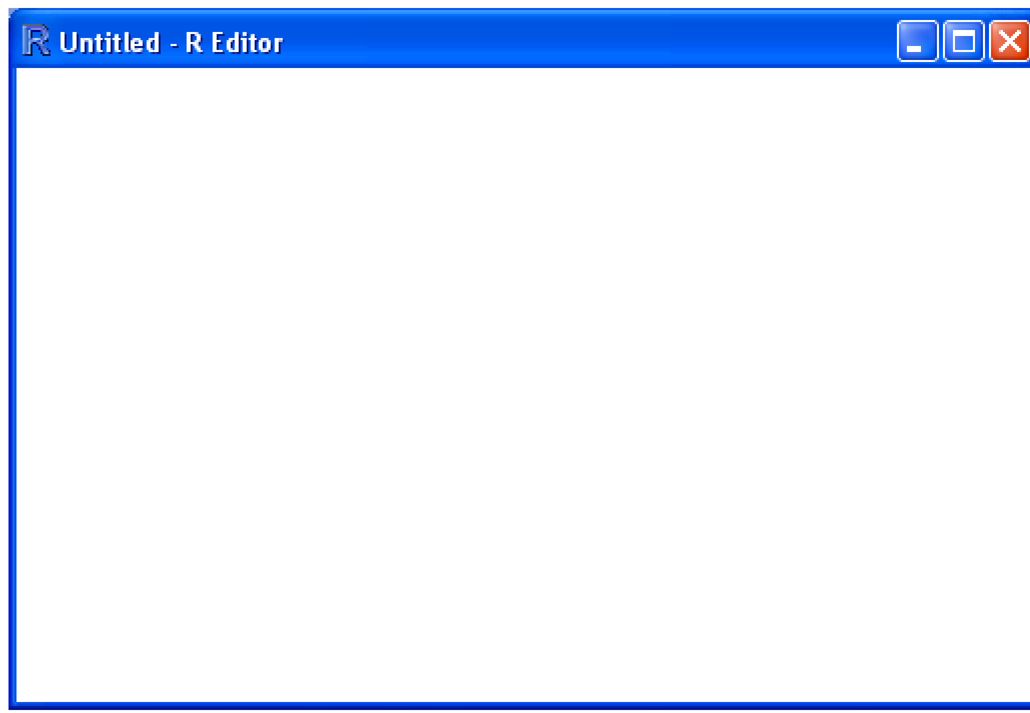
- The R GUI?



Getting Started

- Opening a script.
- This gives you a script window.





R Overview

You can enter commands one at a time at the command prompt (>) or run a set of commands from a source file.

There is a wide variety of data types, including vectors (numerical, character, logical), matrices, dataframes, and lists.

To quit R, use

>q()

R Overview

- Basic assignment and operations.
- Arithmetic Operations:
  - +, -, *, /, ^ are the standard arithmetic operators.
- Matrix Arithmetic.
  - * is element wise multiplication
  - %*% is matrix multiplication
- Assignment
  - To assign a value to a variable use “<-”

R Overview

- If you know which function you want help with simply use ?functionname
- At any time we can list the objects which we have created: ls()
- More commonly a function will operate on an object, for example : sqrt(16)
- Vectors can be created in R in a number of ways. We can describe all of the elements: z<-c(5,9,1,0)

R Overview

- Objects can be removed from the current workspace with the rm function:
- rm(z)
- Sequences can be generated as follows: x<-1:10
- while more general sequences can be generated using the seq command. For example:
- seq(1,9,by=2) or seq(8,20,length=6)

Matrices

- Matrices can be created in R in a variety of ways. Perhaps the simplest is to create the columns and then glue them together with the command cbind.
- > x<-c(5,7,9)
- > y<-c(6,3,4)
- > z<-cbind(x,y)
- > z
- The dimension of a matrix can be checked with the dim command:
- > dim(z)
- Matrices can also be built by explicit construction via the function matrix. For example,
- z<-matrix(c(5,7,9,6,3,4),nrow=3)

R Workspace

Objects that you create during an R session are hold in memory, the collection of objects that you currently have is called the workspace. This workspace is not saved on disk unless you tell R to do so. This means that your objects are lost when you close R and not save the objects, or worse when R or your system crashes on you during a session.

R Workspace

```
# save your command history
    savehistory(file="myfile") # default is ".Rhistory"
# recall your command history
    loadhistory(file="myfile") # default is ".Rhistory“
```

R Datasets

R comes with a number of sample datasets that you can experiment with. Type

> data( )

to see the available datasets. The results will depend on which packages you have loaded. Type

help(*datasetname*)

for details on a sample dataset.

R Packages

– When you download R, already a number (around 30) of packages are downloaded as well. To use a function in an R package, that package has to be attached to the system. When you start R not all of the downloaded packages are attached, only seven packages are attached to the system by default. You can use the function search to see a list of packages that are currently attached to the system, this list is also called the search path.

```
> search()
[1] ".GlobalEnv" "package:stats" "package:graphics"
[4] "package:grDevices" "package:datasets" "package:utils"
[7] "package:methods" "Autoloads" "package:base"
```

“h2o” R package on CRAN

Requirements

- The only requirement to run the “h2o” R package is R >=3.1.0 and Java 7 or later.
- Tested on many versions of Linux, OS X and Windows.

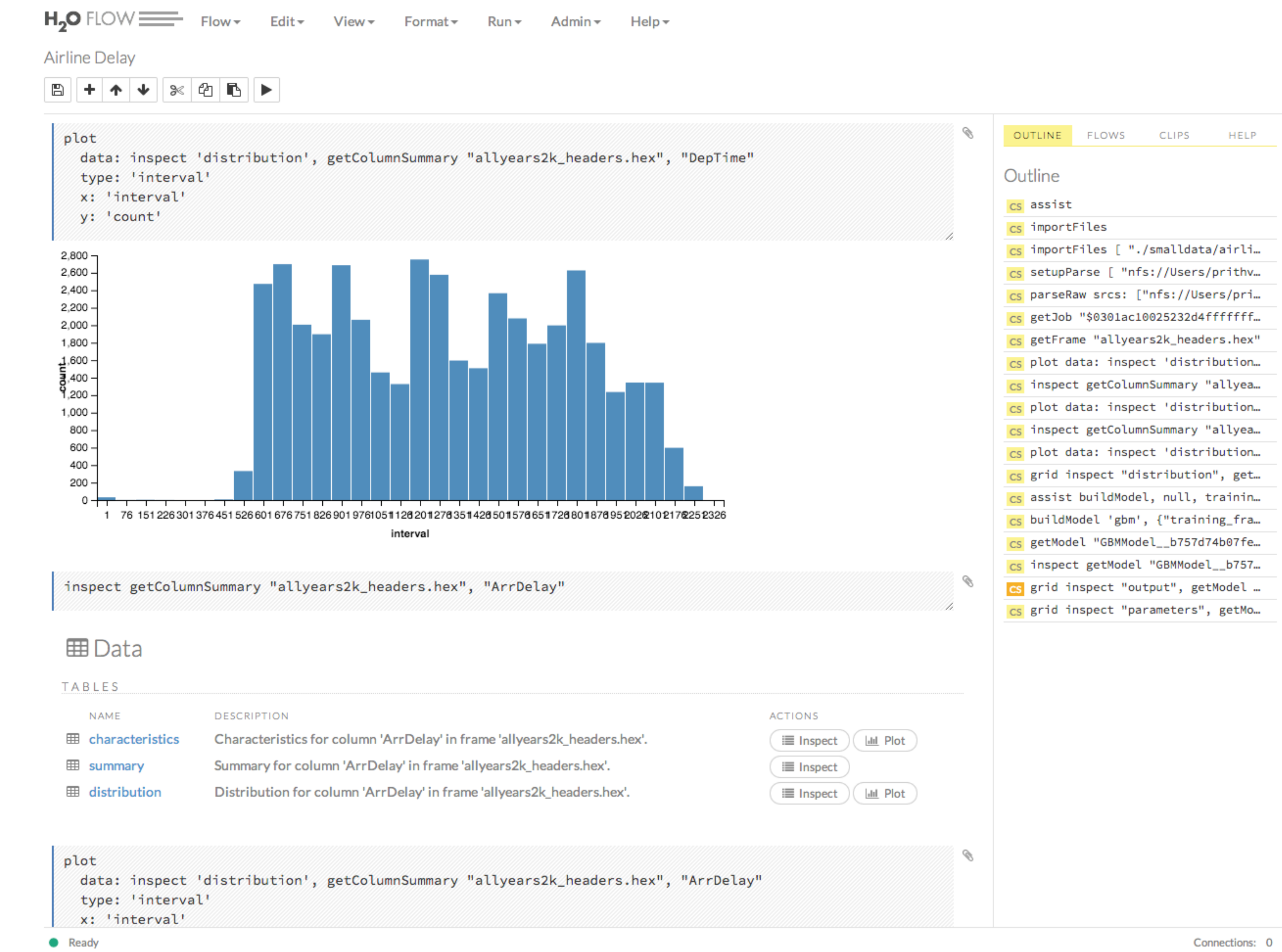
Installation

- The easiest way to install the “h2o” R package is to install directly from CRAN.
- Latest version: http://h2o.ai/download

Design

- No computation is ever performed in R.
- All computations are performed (in highly optimized Java code) in the H2O cluster and initiated by REST calls from R.

H2O Flow Interface



Start H2O Cluster from R

```
> library(h2o)
> localH2O <- h2o.init(nthreads = -1, max_mem_size = "8G")

H2O is not running yet, starting it now...

Note:  In case of errors look at the following log files:
    /var/folders/2j/jg4sl53d5q53tc2_nzm9fz5h0000gn/T//RtmpAXY9gj/h2o_me_started_from_r.out
    /var/folders/2j/jg4sl53d5q53tc2_nzm9fz5h0000gn/T//RtmpAXY9gj/h2o_me_started_from_r.err

java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)

.Successfully connected to http://127.0.0.1:54321/

R is connected to the H2O cluster:
    H2O cluster uptime:         1 seconds 96 milliseconds
    H2O cluster version:        3.3.0.99999
    H2O cluster name:           H2O_started_from_R_me_kfo618
    H2O cluster total nodes:    1
    H2O cluster total memory:   7.11 GB
    H2O cluster total cores:    8
    H2O cluster allowed cores:  8
    H2O cluster healthy:        TRUE

>
```

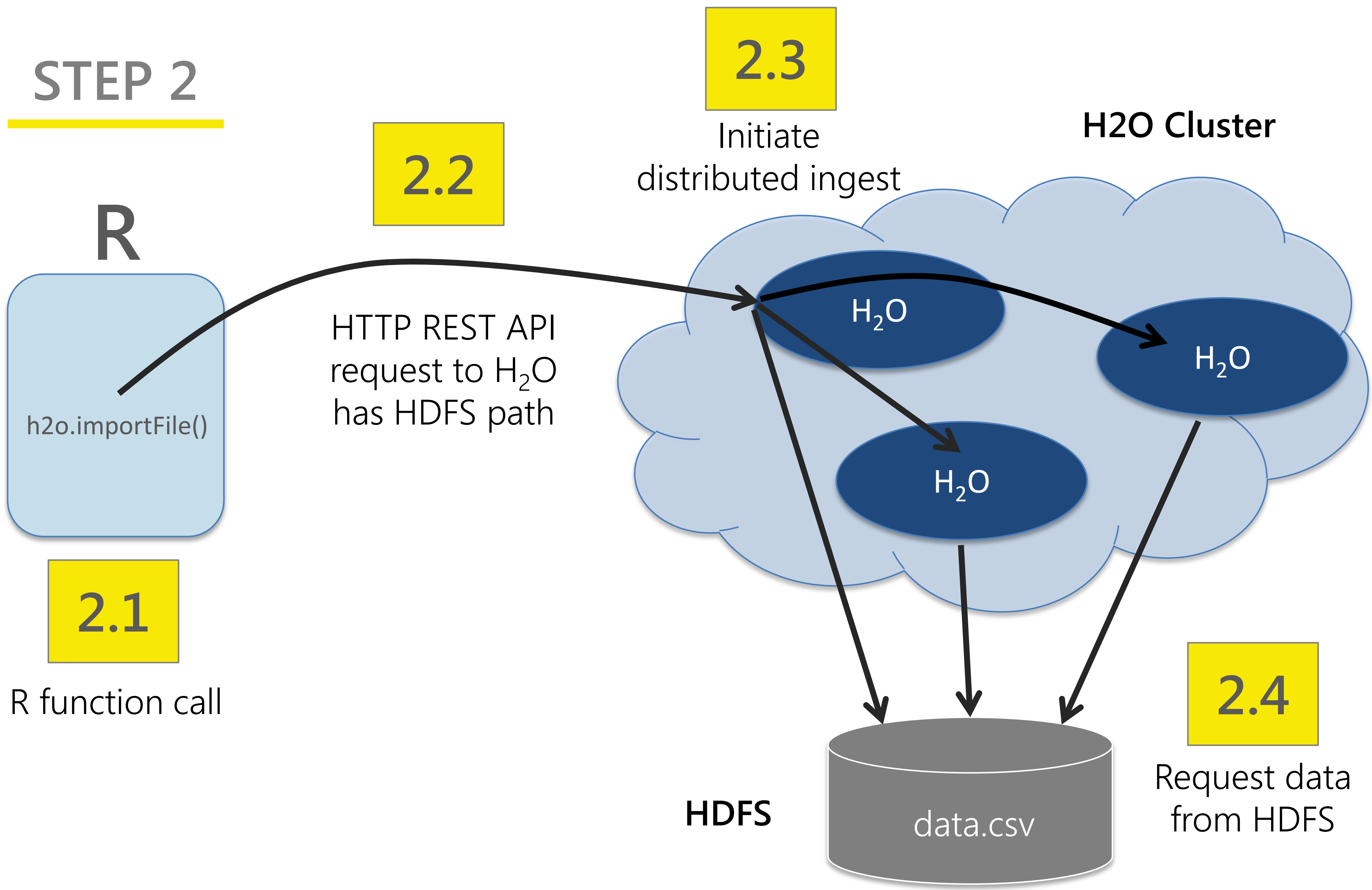
H2O in R: Load Data

Example

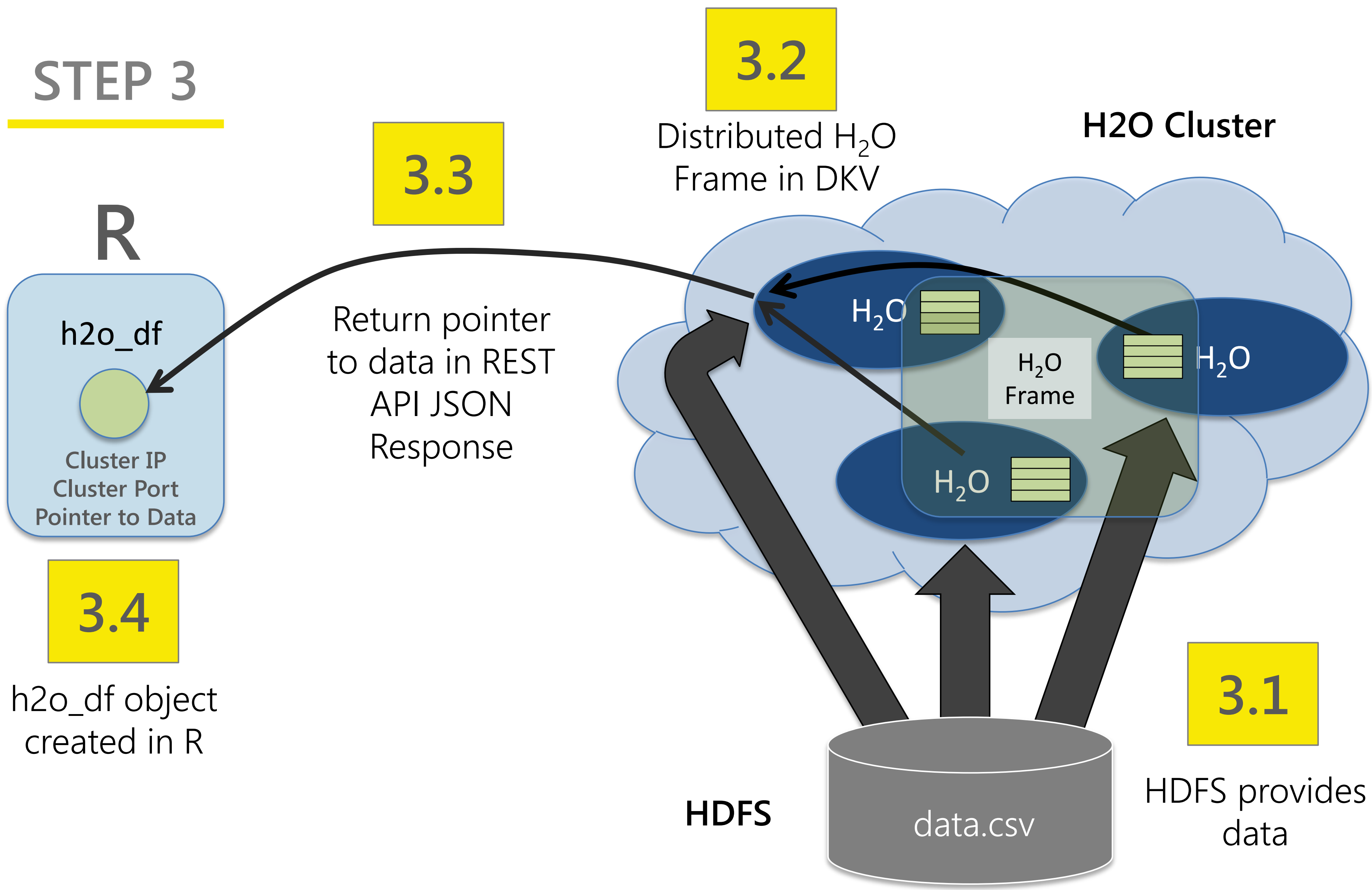
```
library(h2o)  # First install from CRAN
localH2O <- h2o.init()  # Initialize the H2O cluster

# Data directly into H2O cluster (avoids R)
train <- h2o.importFile(path = "train.csv")

# Data into H2O from R data.frame
train <- as.h2o(my_df)
```

R code example: Load data

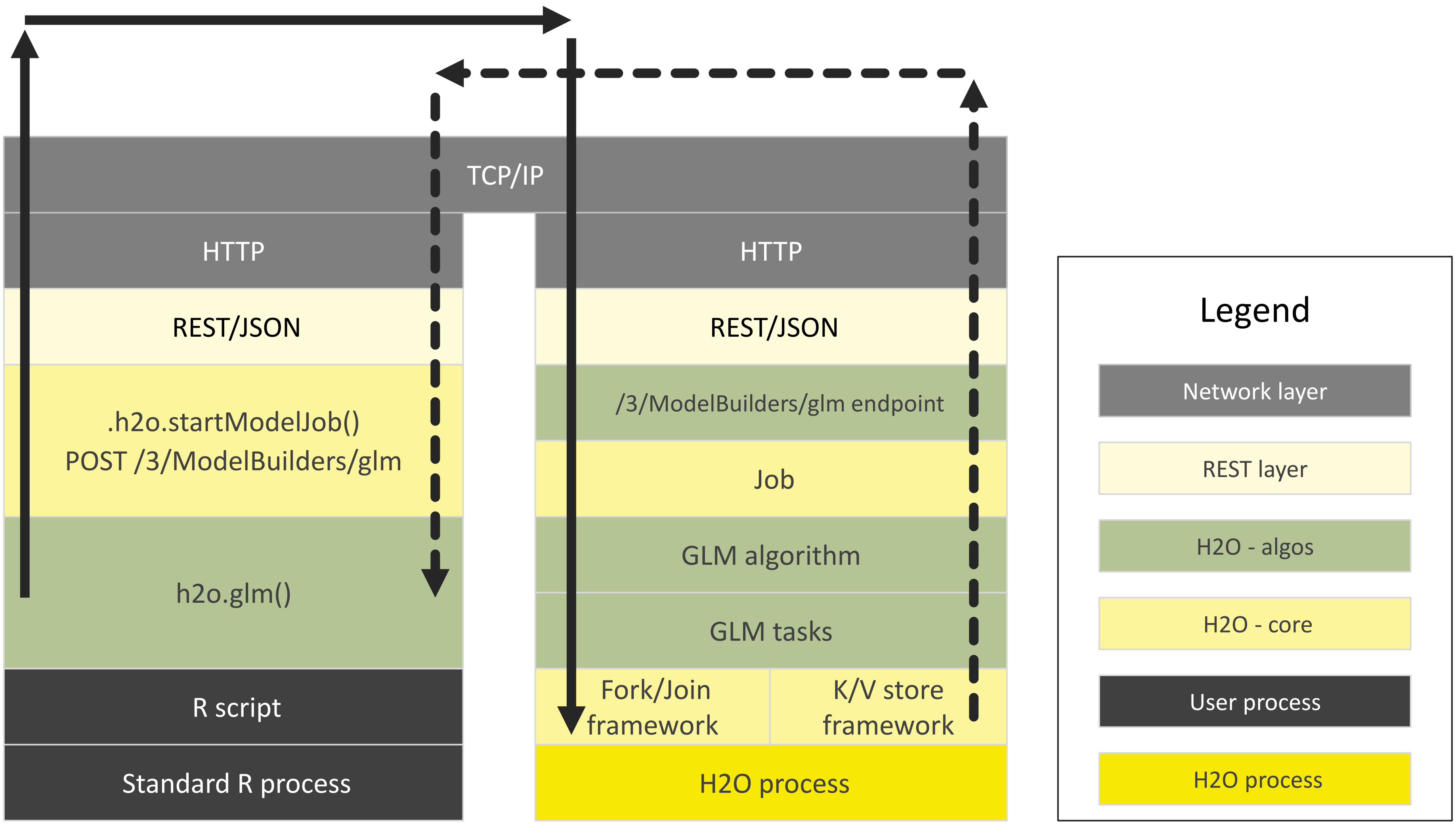
Reading Data from HDFS into H2O with R



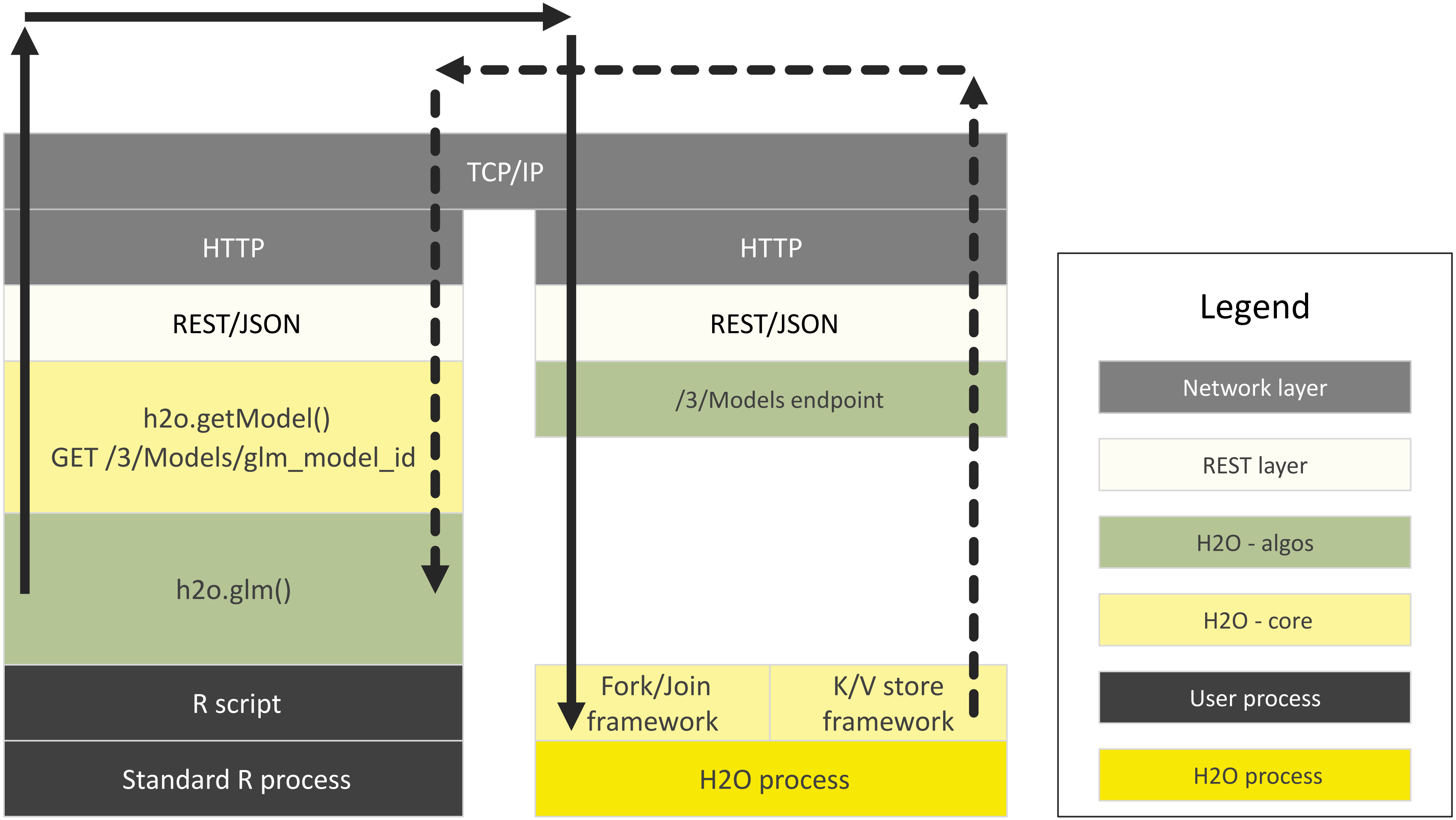
Reading Data from HDFS into H2O with R



R Script Starting H2O GLM



R Script Retrieving H2O GLM Result



H2O Demo!

Thank You