**H2O**

* H2O is an in-memory platform for distributed, scalable machine learning.
* H2O is a Java-based software for data modeling and general computing.
* The primary purpose of H2O is as a distributed (many machines), parallel (many CPUs), in memory (several hundred GBs Xmx) processing engine.
* The **goal of H2O** is to allow simple horizontal scaling to a given problem in order to produce a solution faster. The conceptual paradigm MapReduce (AKA “divide and conquer and combine”), along with a good concurrent application structure, enable this type of scaling in H2O.
* For application developers and data scientists, the gritty details of thread-safety, algorithm parallelism, and node coherence on a network are concealed by simple-to-use REST calls.

**H2O Key Features:**

* Algorithms developed from the ground up for distributed computed and for both supervised and unsupervised approaches including Random Forest, GLM, XGBoost, GLRM, Word2Vec and many more.
* Can use programming language like R or Python to build models in H2O, or use H2O Flow, a graphical notebook based interactive user interface that does not require any coding.
* In-memory processing with fast serialization between nodes and clusters to support massive datasets.Distributed processing on big data delivers speeds up to 100x faster with fine grain parallelism, enabling optimal efficiency without introducing degradation in computational accuracy.

**How H2O works:**

H2O works on existing big data infrastructure, on bare metal or on top of existing Hadoop, Spark or Kubernetes clusters. It can ingest data directly from HDFS, Spark, S3, Azure Data Lake or any other data source into its memory distributed key-value store.

* **Distributed, In-Memory Machine Learning**

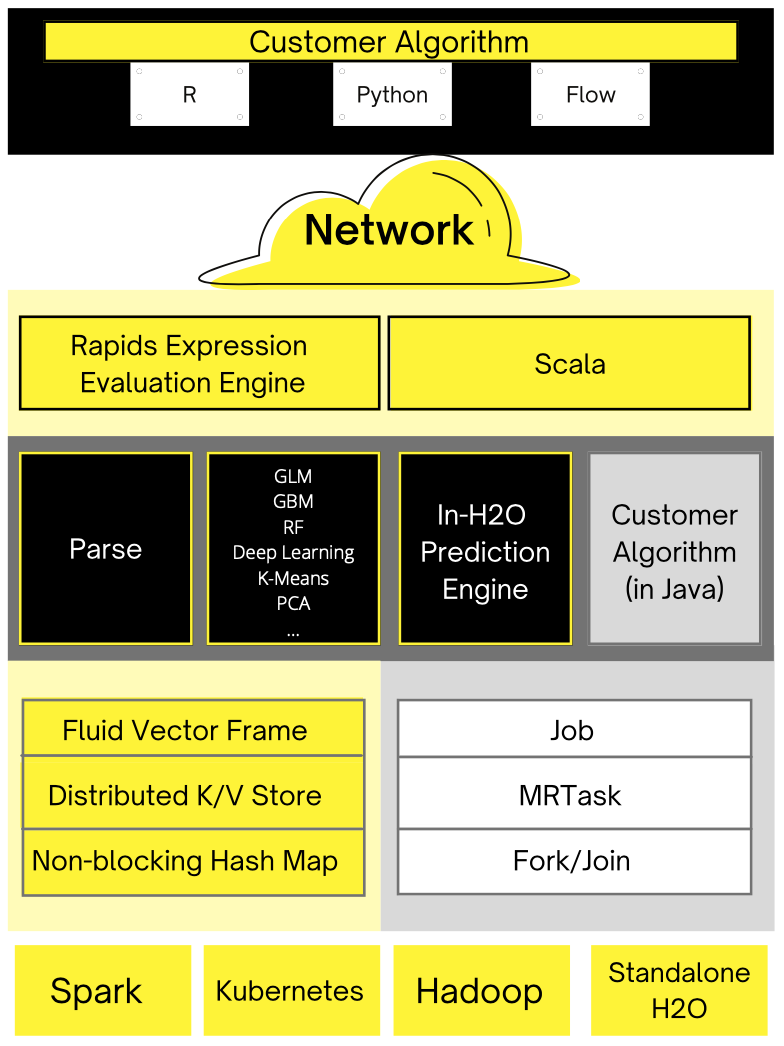
H2O takes advantage of the computing power of distributed systems and in-memory computing to accelerate machine learning using its industry parallelized algorithms which take advantage of fine grained in-memory mapreduce.

* **Seamless Deployment**

Quickly and easily deploy models into production with Java (POJO) and binary formats (MOJO). In addition, H2O models can be productionized in a host of different ways

**H2O Featured Use Cases:**

* Fraud Detection
* Advanced analytics
* Claim management
* Digital advertising

**H2O Architecture**

The diagram shows most of the different components that work together to form the H2O software stack. The diagram is split into a top and bottom section, with the network cloud dividing the two sections.

* The top section shows some of the different REST API clients that exist for H2O.
* The bottom section shows different components that run within an H2O JVM process.
* The color scheme in the diagram shows each layer in a consistent color but always shows user-added customer algorithm code as gray.

**H2O Python Module**

* H2O Python module provides access to the H2O JVM, as well as its extensions, objects, machine-learning algorithms, and modeling support capabilities, such as basic munging and feature generation.
* The H2O JVM provides a web server so that all communication occurs on a socket (specified by an IP address and a port) via a series of REST calls.
* The H2O Python module is not intended as a replacement for other popular machine learning frameworks such as scikit-learn, pylearn2, and their ilk, but is intended to bring H2O to a wider audience of data and machine learning devotees who work exclusively with Python.
* H2O from Python is a tool for rapidly turning over models, doing data munging, and building applications in a fast, scalable environment without any of the mental anguish about parallelism and distribution of work.

**Who Uses H2O & Why?**

* A number of well-known companies are using H2O for their big data processing, and the website claims that over 5000 organizations currently use it.
* The company behind it, H2O.ai, has over 80 staff, more than half of which are developers.
* The primary things H2O brings are ease of use and efficient scalability to data sets too large to fit in the memory of your largest machine.
* For SparkML users, who feel they already have that, H2O algorithms are fewer in number but apparently significantly quicker.
* As a bonus, the intelligent defaults mean your code is very compact and clear to read: you can literally get a well-tuned, state-of-the-art, deep learning model as a one-liner.

**How H2O is used?**

H2O is in its core a platform for distributed, in-memory computing. On top of the distributed computation platform, the Machine learning algorithms are implemented. At H2O, we design every operation, be it data transformation, training of machine learning models or even parsing to utilize the distributed computation model. In order to work with big data fast, it’s necessary. However, a single operation usually cannot utilize cluster’s computational resources to the very maximum. Data need to be distributed across the cluster, many operations require sequential execution of tasks, which, even if implemented in a distributed manner, follow after each other and require data exchange. These and many other smaller factors, if summed up together, may introduce a significant overhead.

* H2O is software for machine learning and data analysis.
* H2O is:
  1. Open source (the liberal Apache license)
  2. Easy to use
  3. Scalable to big data
  4. Well-documented and commercially supported
  5. On its third version (i.e., a mature architecture)
  6. Wide range of OS/language support
* H2O retains the attitude of not just “How do we get this to work?” but “How do we get this to work efficiently at big data scale?” permeating the whole development.
* If machine learning has come of age, H2O looks to be not just an economical family car for it, but simultaneously the large load delivery truck for it.
* H2O uses familiar interfaces like R, Python, Scala, Java, JSON and the Flow notebook/web interface, and works seamlessly with big data technologies like Hadoop and Spark.
* H2O provides implementations of many popular algorithms such as Generalized Linear Models (GLM), Gradient Boosting Machines (including XGBoost), Random Forests, Deep Neural Networks, Stacked Ensembles, Naive Bayes, Generalized Additive Models (GAM), Cox Proportional Hazards, K-Means, PCA, Word2Vec, as well as a fully automatic machine learning algorithm (H2O AutoML).
* H2O is extensible so that developers can add data transformations and custom algorithms of their choice and access them through all of those clients.
* As a data scientist, one is most likely to use R and/or Python. H2O integrates with both. Interestingly, H2O makes it easy to seamlessly switch Python, R, and other data science tools while still working on the same project. This allows data scientists to interact more easily, as well as use the best tool for the job.
* The possibilities do not stop there. H2O also offers its own web-based interface named Flow. By means of Flow, data scientists are able to import, explore, and modify datasets, play with models, verify models performances, and much more. Flow is beautiful and a quick way to do machine learning. Flows can be saved and given to other data scientists, making cooperation easy.

**Where H2O is used?**

**Anomaly detection** : Anomaly detection is a common data science problem where the goal is to identify odd or suspicious observations, events, or items in our data that might be indicative of some issues in our data collection process (such as broken sensors, typos in collected forms, etc.) or unexpected events like security breaches, server failures, and so on. Anomaly detection can be performed in a supervised, semi-supervised, and unsupervised manner.

**Grid Search** : Grid search “walks” through a space of hyperparameters and builds the respective models. The walking strategy might differ, but they all share the same problem - there are many models to be built. Typically, the build process is constrained by time. And the more models are built, the better.Since H2O had given users a way of building models using Grid Search in parallel. This effectively means there may be n > 1 models trained on the cluster. While one model is waiting or doing less optimizable calculations, other model can meanwhile utilize cluster’s resources. This leads to **more models built in less time.**

**AutoML:** AutoML interface is designed to have as few parameters as possible so that all the user needs to do is point to their dataset, identify the response column and optionally specify a time constraint or limit on the number of total models trained.

**Deep Learning**: Deep learning with neural networks is arguably one of the most rapidly growing applications of machine learning and AI today. They allow building complex models that consist of multiple hidden layers within artifiical networks and are able to find non-linear patterns in unstructured data. Deep neural networks are usually feed-forward, which means that each layer feeds its output to subsequent layers, but recurrent or feed-back neural networks can also be built. Feed-forward neural networks are also called multilayer perceptrons (MLPs). The R package h2o provides a convenient interface to H2O, which is an open-source machine learning and deep learning platform. H2O can be integrated with Apache Spark **(**[**Sparkling Water**](http://www.h2o.ai/sparkling-water/)**)** and therefore allows the implementation of complex or big models in a fast and scalable manner. H2O distributes a wide range of common machine learning algorithms for classification, regression and deep learning.

**Free Options Available:**

* **H2O Wave:** H2O Wave is an open-source Python development framework that makes it fast and easy for data scientists, machine learning engineers, and software developers to develop real-time interactive AI apps with sophisticated visualizations. H2O Wave accelerates development with a wide variety of user-interface components and charts, including dashboard templates, dialogs, themes, widgets etc. H2O Wave’s real-time application server enables developers to stream on-going changes to dashboards, model results, and any other dynamic information in the application.
* **H2O AutoML:** AutoML or Automatic Machine Learning is the process of automating algorithm selection, feature generation, hyperparameter tuning, iterative modeling, and model assessment. AutoML make it easy to train and evaluate machine learning models. Automating repetitive tasks allows people to focus on the data and the business problems they are trying to solve.
  1. Train the best model in the least amount of time to save human errors.
  2. Reduce the need for expertise in machine learning by reducing the manual code-writing time.
  3. Improve the performance of machine learning models.
  4. Increase reproducibility and establish a baseline for scientific research or applications.
  5. Scales training dataset to clusters (Hadoop, Spark, Kubernetes)

Following are the aspects of AutoML:

* 1. Imputation, one-hot encoding, standardization
  2. Feature selection and/or feature extraction
  3. Count/Label/Target encoding of Categorical Features
  4. Cartesian Grid search or Random Grid search
  5. Bayesian hyperparameter optimization
  6. Individual models can be tuned using a validation set
  7. Ensembles often out-perform individual models
  8. Stacking/Super Learning(Wolpert, Breiman)
  9. Ensemble Selection(Caruana)