Technical Report

On

Analysis of Machine Learning Library (H2O vs SparkMLlib)

Saurabh Chauhan

Contents

1	**Better Machine Learning with H2O**	**1**
2	**End-to-End Machine Learning with SparkMLlib**	**2**
3	**Confused with H2O and SparkMLlib ?**	**2**
4	**How to choose Machine Learning Algorithms**	**3**
5	**H2O integration with SparkR**	**4**
6	**Experimental Results**	**4**
7	**Conclusions**	**7**

List of Figures

1 Predictive Analysis using H2O in SparkR 5
2 Predictive Analysis using SparkMLlib in SparkR 6

List of Tables

1 Score Models with Confidence 2
2 H2O and SparkMLlib supported Machine Learning Algorithms 2
3 Comparison: Predictive Analysis algorithm using H2O and SparkMLlib 6

1 Better Machine Learning with H2O

H2O[1] is open-source software from open-source community for big-data analysis targeting big-data community. It enables the rapid solution of business problems by applying machine learning algorithms. It provides following unique features to enable rapid application of machine learning applications:

- **Open Source Technology** - It provides the freedom enabling machine learning and predictive analysis offered by open source community. It offers flexibility by providing the integration with most popular big-data technology Apache Hadoop and Spark etc.
- **User-friendly Interfaces** - H2O embedded with the web-based graphical user interface and programming interfaces with familiar environments like R, Scala, Java, Python etc.
- **Data Agnostic Support** - H2O provides data agnostic support with HDFS, S3, SQL, No-SQL including all common database and file types. It also explore and model data through R Studio, Tableau, MS Excel etc.
- **State of the Art Machine Learning Algorithms** - H2O available with bunch of machine learning algorithms ranging from Generalized Linear Model (GLM), Random Forest, Gradient Boosting (GBM), K-Means, Anomaly Detection, Deep Learning to Naive Bayes to solve the business problems.
- **Better Pre-processing** - H2O equipped with following munge tools:
  - Data Profiling - Summarize the shape of the dataset including missing values, zero values upon data ingestion.
  - Summary Statistics - Visualize the data with statistical measure including mean, min-max, standard deviation, quartile and provides the preview of the dataset.
  - Aggregate Column - Provides unique views with binning, filtering, group functions, and derived columns.
  - Transformation - Normalize, transform, and partition the data to fit into the model by providing the right shape.
  - PCA - It simplify and speed-up feature selection process with simple to use interface.
  - Training and Validation Sampling - Provides random sampling plan to generate training test for parameter estimation and validation set for model validation.
- **Score Models with Confidence** - Output of any model can be described using following model metrics which defines goodness of fit for the model.

Thus by combining power of advanced machine learning and predictive analysis algorithms, the truly open source freedom, scalable in-memory processing for big data make it faster, easier and cost-effective for business problems.

[1] http://www.h2o.ai/

Model Metrics	Description
Predict	Predict the outcome of a data set with any model (GLM, GBM, Decision Tree and Deep Learning Models etc.)
Confusion Matrix	Tabular representation to understand how a model performs.
AUC	A graphical plot to visual the performance of a model using true positive, and false positive.
Hit Ratio	A classification matrix to measure the ratio of the number of correctly classified and incorrectly classified cases.

Table 1: Score Models with Confidence

Algorithms	H2O	SparkMLlib
Supervised Leaning	Generalized Linear Model, Gradient-Boosting, Random Forest, Naive Bayes, Deep Leaning	Naive Bayes, Generalized Liner Regression, Survival Regression, Decision Tree, Random Forest, Gradient Boosting, Alternating Least Squares,
Unsupervised Learning	K-Means Clustering, Principal-Component Analysis	K-Means, Gaussian Mixtures

Table 2: H2O and SparkMLlib supported Machine Learning Algorithms

2 End-to-End Machine Learning with SparkMLlib

MLlib[2] is spark's machine learning (ML) library with the goal of enabling practical machine learning easy and scalable to derive the business solutions. In an abstract manner, ML is divided into following two packages:

- SPARK.MLLIB - Provides the original API built on top of RDDs.
- SPARK.ML - Contains higher-level API built on top of DataFrames.

From above two packages, SPARK.ML is recommended because it is built on top of DataFrames and with DataFrame the API is more flexible and adapted by many different functions. The SparkMLlib also offers the wide range of machine learning algorithms ranging from Linear Models (logistic regression, linear regression), Naive Bayes, Decision Trees, Clustering (K-means), Dimensionality Reduction (SVD, PCA), to Frequent Pattern Mining (FP- growth, association rules).

3 Confused with H2O and SparkMLlib ?

As explained in the section 2, SparkML is different from SparkMLlib. SparkML operates on the DataFrame which is new in the Spark (introduced in Spark 2.0) however, it doesn't allow all types of algorithms. In general use of DataFrame (SparkML) is faster than RDD (SparkMLlib). When we are in the initial stage and not able to decide which algorithm is best, than SparkMLlib may be suitable (a lot more users use SparkMLlib). But if we are from the algorithmic background and

[2]https://spark.apache.org/docs/2.0.0-preview/mllib-guide.html

looking to tune up a model for speed up than H2O is the best option for sure.
The benchmarking results[3] presented by Dr. Szilard state that:

- The `H2O` algorithms for *linear models* are most memory efficient, and fastest compare to `SparkMLlib` linear models.

- *Random Forest* implementation of `H2O` is more fast and memory efficient as it uses all available cores. While `SparkMLlib` is a slower with larger memory footprint as it doesn't handle categorical variables automatically in contrast with `H2O`. To implement Gradient Boosting Machine (GBM) algorithm, `H2O` is also fastest as compare to `SparkMLlib` because of multi-threaded architecture.

4 How to choose Machine Learning Algorithms

The list of H2O and SparkMLlib supported ML algorithms are listed in Table 2 and that can be applied in following scenarios:

- *Linear Regression*: When the goal is to predict/estimate the real values (total sales, sale quarter wise, rainfall value, cost of house etc.) based on continuous variables, then linear regression establishes the relationship between dependent and independent variables.

- *Logistic Regression*: It is a classification algorithm used to predict the value of binary variables (0/1, yes/no, true/false etc.). The most widely used of logistic regression are to classify whether an email is spam or not spam, the image contains face or not etc.

- *Gradient Boosting*: GBM is used when we deal with a large amount of data to make the prediction with high prediction power (accuracy). It is the extension of linear and non-linear regression models by combining the prediction of several estimators in order to build the strong predictor.

- *SVM (Support Vector Machine)*: It is also classification method used to classify data points in n-dimensional space with the large margin.

- *Naive Bayes*: It is classification technique based on Bayes' theorem extending the capability of Logistic Regression. Some of the real world example include classification of a news article about technology, sports, or entertainment, and checking text with positive or negative emotions etc.

- *Random Forest*: It gives tree like structure with various classification rules. For instance, if we want to classify the playing activity (i.e. playing ball) based upon the weather conditions than random forest (decision tree) determines the rule that of the weather is sunny and humidity is less than 70, then it's probably OK to play.

- *Deep Learning*: It is a branch of machine learning and attempts to model high-level abstractions in data using multiple linear and non-linear transformations. The real world examples of deep learning include automatic object classification in photographs, handwriting generation, colorization of black and white images etc.

[3] https://github.com/szilard/benchm-ml

- *K-Means*: It is the unsupervised form of learning which solves the clustering problem. It can be used in the scenario where we want to group persons based on their nature, group of similar items in a close region in the shopping mall, group of the asian country etc.
- *PCA -Principal Component Analysis*: In the last few years, data is generated at the exponential rate. For example, e-commerce companies are capturing detailed information about customers including their demographics, like/dislike, purchase history/pattern, many other details to identify the nature of the person using above features. While model development some of the parameters are not important (not contributing to model development) such parameters can be omitted using PCA to enable robust model development over the large volume of versatile data.

5 H2O integration with SparkR

The prerequisites to integrate H2O with SparkR are:

- 64-bit Java version 1.6 or newer
- R version 2.13 or newer
- Spark release 2.0.0

Apart from above prerequisites following steps need to be performed to integrate H2O with existing SparkR version:

- Open SparkR console either through terminal or GUI provided by RStudio
- Integrate Spark with RStudio by installing `library(SparkR)` library and point `Sys.setenv` to `SPARK_HOME` (required if using RStudio for integration).
- Initiate the instance of SparkR using `sparkR.session()` (required if using RStudio)
- Install H2O for package for SparkR: `install.packages(h2o)`
- Run the H2O cluster on top of Spark using `h2o.init()`

6 Experimental Results

We have implemented predictive analysis algorithm using `H2O` as well as using `SparkMLlib`. The experiment is performed on the 64-bit system having 4.00 GB RAM with Core-i3 2.40 GHz processor. Additionally, predictive analysis is carried out on iris data set (inbuilt data set with size of 7088 bytes) and predicted value of SEPAL_LENGTH variable using `h2o.glm` and `spark.glm` function. We have also performed the experiment on the airlines [4] data set (with size of 53688 bytes) and predicted value of ISARRDELAYED (binary variable) presented in Table 3.

- **Predictive analysis using H2O** - We have implemented above described scenario (predictive analysis on iris data set) using `h2o` library in SparkR. We are initiating the H2O cluster on top of SparkR using `h2o.init()`function, reading H2O data frame using `as.h2o()` function, and generating training and validation sets using `h2o.splitFrame()` function followed

[4]Data set is available at https://github.com/h2oai/h2o-2/wiki/Hacking-Airline-DataSet-with-H2O

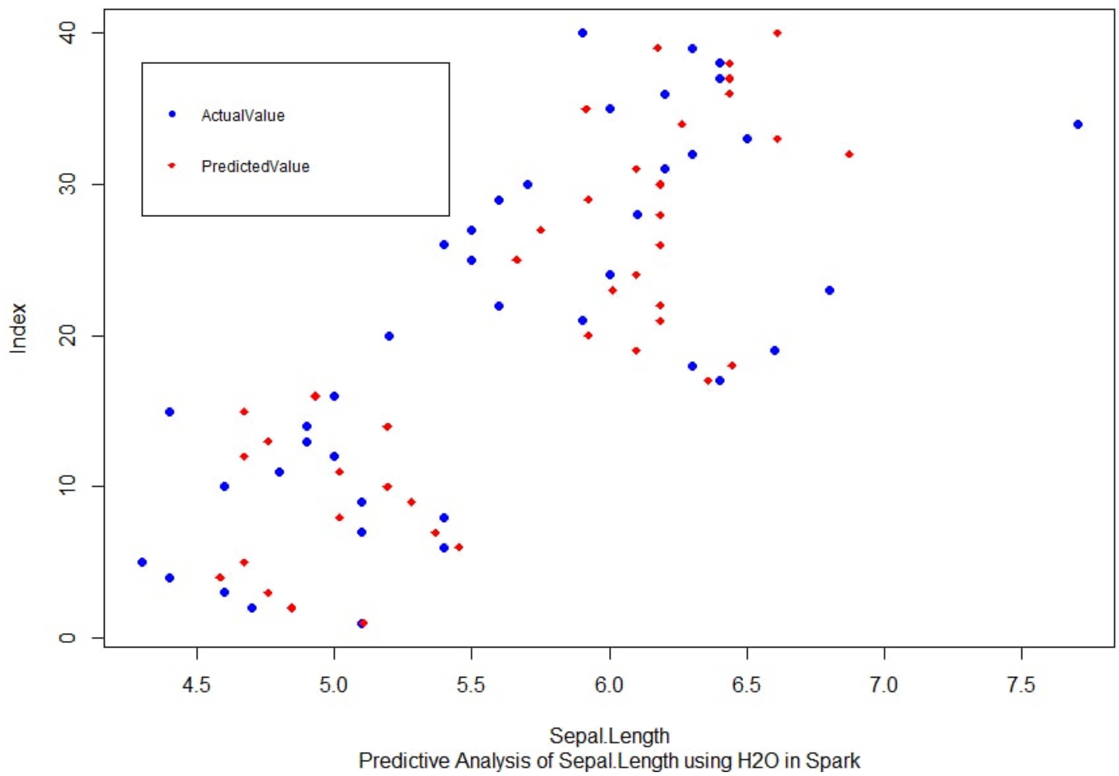

Figure 1: Predictive Analysis using H2O in SparkR

by predictive analysis using `h2o.glm()` over the training set. The measurement metric is available through `summary()` function, which describes the information about the goodness of fitted model. Finally validation can be done using `h2o.predict()` function.
The time taken by above algorithm is 15.86 secs starting from SparkR and H2O initialization, followed by model development to plotting actual values vs expected values. The accuracy (R^2) of the used model is 71.33 % with error (RMSE) 0.43. The following graph (Fig 1) shows the actual values (blue dots) and predicted values (red dots) calculated using the H2O predictive model.

- **Predictive analysis using SparkMLlib** - We have implemented the same scenario using `sparkR` library in R. We are initiating the spark environment on top of R using `sparkR.session()` function, reading spark data frame using `as.DataFrame()` function, and generating training and validation sets using `caTools` package followed by predictive analysis using `spark.glm()` over the training set. The `SparkMLlib` doesn't provide detailed/required measurement metric to determine the goodness of fitted model (measurement metric is calculated using statistics formula). Finally, validation can be done using `predict()` function. **The time taken by above algorithm is 25.70 secs starting from SparkR initialization, followed by model development to plotting actual values vs expected values. The accuracy (R^2) of the used model is 68.33 % with error (RMSE) 0.47**. The following graph (Fig 2) shows the actual values (blue dots) and predicted values (red dots) calculated using the `SparkRMLlib` predictive model.

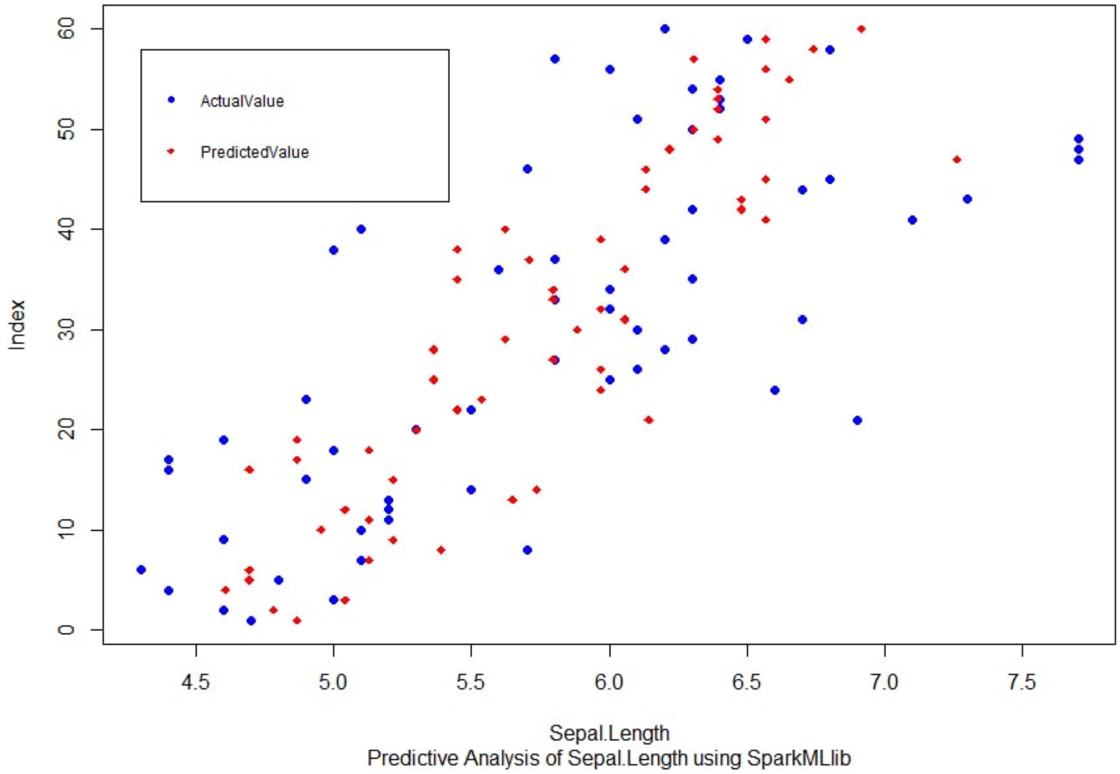

Figure 2: Predictive Analysis using SparkMLlib in SparkR

- **Comparative Analysis** - The comparative analysis results are presented in the Table 3. The results represent that algorithm implemented using `H2O` is more efficient (in terms of R^2 and RMSE-root mean square error) compare to the algorithm implemented using `SparkMLlib`. Apart from the efficiency, `H2O` library is quick (in terms of execution time) compare to `SparkMLlib` as presented in Table 3. The reason for the quick response of `H2O` is, it provides the function to partition the data into training and validation set while in the case of `SparkMLlib` this is missing. So, we need to convert Spark DataFrame to R/H2O data frame and back to Spark DataFrame. This conversion introduces additional overhead. For the large dataset, Spark crashes the system because it's larger memory footprint and it doesn't provide the mechanism to encode the categorical variables (therefore most the users use R to encode categorical variables).

Library for Predictive Analysis	R^2 (efficiency)	RMSE	Execution Time (in secs)	Distribution Family	Data Size (bytes)
H2O	**71.33 %**	**0.43**	**15.86**	**Gaussian**	**7088**
SparkMLlib	**68.33 %**	**0.47**	**25.70**		
H2O	**91.54 %**	**0.15**	**127.2**	**Binomial**	**53688**
SparkMLlib	**System crash**				

Table 3: Comparison: Predictive Analysis algorithm using H2O and SparkMLlib

7 Conclusions

- The benchmarking results state that H2O is much faster, accurate than SparkMLlib in the various machine learning algorithms starting from linear models, random forest, gradient machines, to deep neural networks.
- It also performs better with the data set having 10K records to 10M records.
- In addition to benchmarking result, experimental results highlight the features of H2O machine learning algorithm over SparkMLlib (machine learning) library.
- It's not a huge leap to move from H2O machine learning library to SparkMLlib (if required).