УДК 004.24:004.6

FACTORS AFFECTING MACHINE LEARNING ALGORITHMS SELECTION

B. ZIBITSKER, MS, PhD
and Honorable Doctor at BSUIR
President and CEO BEZNext,
Adjunct Associate Professor, DePaul University in Chicago

D. A. HEGER, PhD
Founder of DHTechnologies, Data Analytica, Hot-shot Analytics and AI/ML firm, Austin, USA

CEO BEZNext, Chicago, USA
DHTechnologies, Austin, TX, USA
E-mail: bzibitsker@beznext.com

Abstract. Selection of the Machine Learning (ML) algorithms and ML Libraries affect accuracy, response time, scalability and success of implementing new Big Data applications. Unfortunately, algorithms providing high accuracy not necessarily provide good response time and scale well. Different algorithms take different training time and different efforts for operationalization. In this paper we will discuss results of collaborative efforts on benchmarking ML algorithms and libraries and review the algorithm of recommender selecting the appropriate ML algorithm and ML library for new Big Data applications, depending on relative importance of accuracy, response time, scalability and other criteria.

Key words: Performance Assurance, Performance Engineering, Benchmarking, Machine Learning Algorithms and Machine Learning Benchmark, Machine Learning Algorithm Selection and Machine Learning Library Selection

1 Introduction.

After selection the class of the ML algorithms for solving of the business problem Data Scientists face a challenge of selecting the appropriate type of the algorithm and ML library providing sufficient accuracy, response time, scalability, minimize usage of resources and cost [1].

Performance of ML algorithms depends on many factors including data set sizes and number of predictors/attributes

When data set size and number of predictors / attributes is small the best results can be obtained with one type of the algorithms and when data set is large another algorithm can provide better results.

We will take it into consideration while developing Recommender for selecting appropriate ML algorithm and ML library.

In this paper we review measurement data collected during benchmark tests of the several ML algorithms. We ran Python programs incorporating Regression algorithms accessing the data sets with different number of observations and predictors/attributes. Collected measurement data include response time, throughput, accuracy, CPU utilization, number of I/O operations, memory utilization and network utilization.

For the small data sets a standalone Python program outperformed the Python program running in Spark environment. Indeed, for small data sets the overhead of Spark does not compensate the savings due to parallel processing.

We determined the minimum size of the data set when parallel processing savings compensate the overhead of Spark.

Measurement data were used to build ML and QNM models, expanding results of the benchmark tests.

The Recommender use the modeling results and business requirements to accuracy and performance to select appropriate ML algorithm and library.

2 Role of Benchmarks

We created a methodology of conducting benchmarks to evaluate performance and accuracy of the different ML algorithms and libraries and analyze the impact of the size of the data set on time of the models training.

Benchmarking process shown on Figure 1. includes the following steps:

Preparing the Data Sets with different numbers of observations and predictors

1 Preparing and running the Benchmark test:

– Writing Python programs
– Creating the benchmark environment
– Collecting measurement data about response time, accuracy, CPU, memory usage, I/O rate and network utilization.

2 Modeling:

– Building models to predict performance characteristics of different ML algorithms and ML Libraries for different sizes of the data sets not included into the benchmarks

3 Model validating:

– Comparing the prediction results with actual measurement data for data set with different number of observations and predictors.

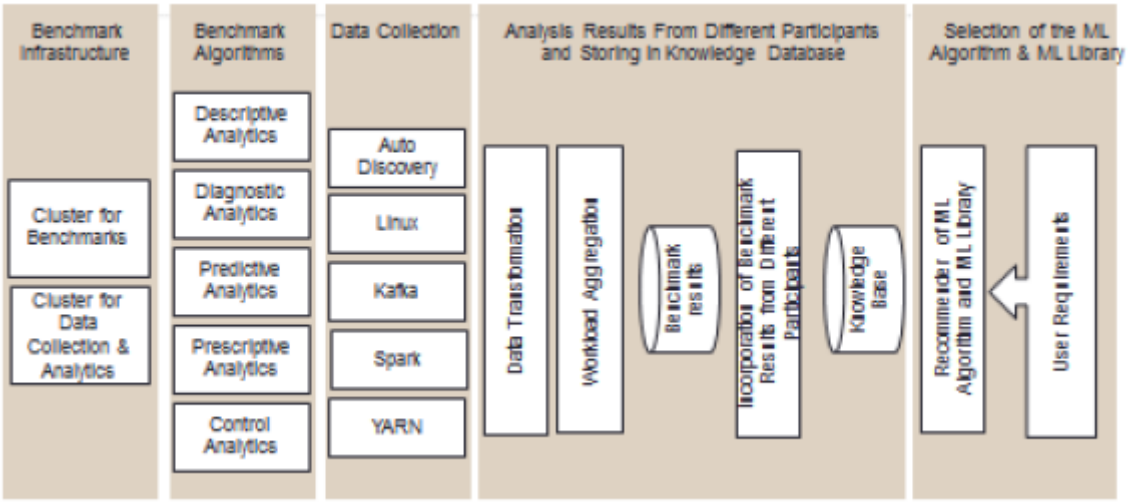

Figure 1. Benchmark Process for testing Python programs ran standalone and in Spark environment

Infrastructure for the benchmark tests of ML algorithms and libraries provided by IBM includes 4 Data nodes Spark Cluster and 4 Control nodes used for management, data collection, and analysis.

Benchmark tests of the different ML algorithms are performed by different collaborators, including Universities and Research organizations. Results of the benchmark tests performed by collaborators will be transformed to common format and stored in common repository.

In first benchmark test we ran OLS (Figure 2.), Ridge, RF (Figure 3.) algorithms implemented in Python as standalone application and as Spark Application.

The second benchmark test ran against Higgs Boson Data Set consisting of 11,000,000 instances and 28 predictors/attributes. We will review the results of testing several ML algorithms on various data set sizes.

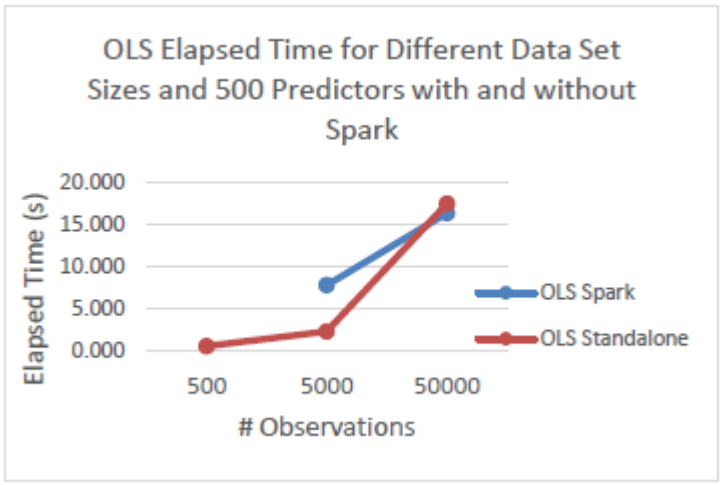

Figure 2. Measurement data collected during benchmarking Regression algorithm's OLS. Graph shows change of the elapsed time for 3 different data sets sizes with 500 predictors for Python program running in standalone and Spark environment. Only when data set size exceeds 50,000 observations parallel processing in Spark provides the performance advantage

Each algorithm was tested for 22 dataset sizes with number of predictors changing from 2 to 500 incremented by 50 with 500 and 5000 observations.

Each benchmark test was run for 2 hours.

Measurement data include Accuracy, Response time, CPU utilization, Memory usage, I/O rate and Network throughput changed during the benchmark tests with the increase of the number of observations and number of predictors.

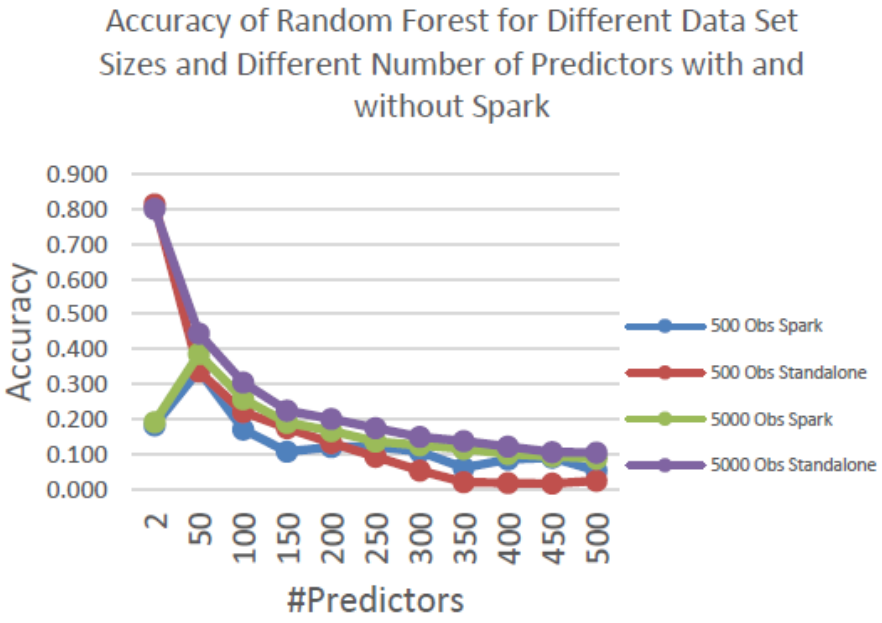

Figure 3. The accuracy of the Random Forest algorithms for different sizes of the data set and number of predictors/attributes changing between 2 and 500 for Python program running in standalone and Spark environments

The second benchmark was run against Higgs Boson Data Set consisting of 11,000,000 instances and 28 predictors/attributes. Below are the results of testing several ML algorithms on various

data set sizes. In this case we reviewed the data size impact on training and running time. A Linux cluster was used for the analysis.

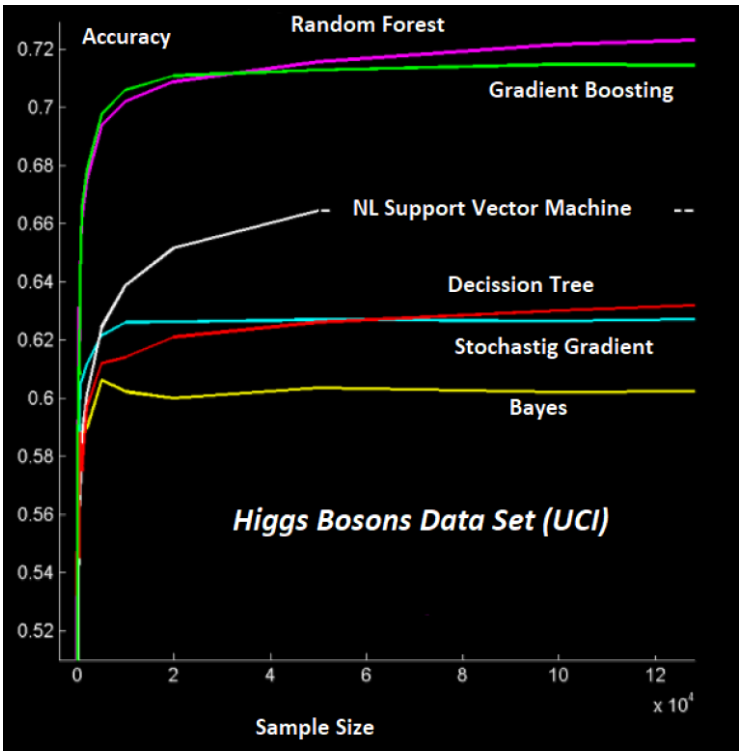

Figure 4. Accuracy of the several ML algorithms depending on Data Set size

In general, every ML algorithm eventually leads to a plateau based on using a subset of the available data set. One cannot upfront stipulate though what the effective number of data points will be for reaching the plateau (where the accuracy does not improve anymore). This is due to the fact that every dataset that is being analyzed is unique and hence for some datasets, even after 100ds of thousands of data points are processed, the accuracy of a certain algorithm being used can still be improved while with others, after processing a few 1000 data points, the plateau is reached.

However, one can always differentiate between 2 broad forms of algorithms, namely parametric (such as linear algorithms) and non-parametric (as an example a random forest or a NL SVM) models. So, in the case a plateau is reached with a non-parametric algorithm, the remaining data points in the dataset are not adding anything to the accuracy and so can be ignored. To illustrate, In Figure 4, the stochastic gradient method reaches a plateau rather quickly (small input dataset size) and just flattens out on the accuracy scale. That basically implies that based on the dataset at hand, this is the best this algorithm can do! Hence, non-parametric algorithms normally work well on complex models (like deep-learning) as they can really benefit from a vast number of training samples. To reiterate, it all depends on the content and the size of the dataset that has to be mined.

It has to be stressed here as well that optimizing the parameters of the algorithms has a profound impact on the speed of convergence to a plateau - where cross-validation has to be part of the equation. Another advantage of what is being discussed here is that if only a subset of the data points may be needed to get to the plateau (from an accuracy perspective), as less data points are being used, less memory and CPU cycles are being consumed and so the run-time behavior of the algorithms can be optimized rather significantly.

In general, the more potent machine learning algorithms are often referred to as nonlinear algorithms. By definition, they are able to learn complex nonlinear relationships among the input and output features. So, these algorithms are normally more flexible and even non-parametric (aka they

can determine how many parameters are required to model a problem in addition to the values of those parameters). These algorithms are also high-variance, meaning that predictions vary based on the specific data used to train them. This added flexibility and power comes at the cost of requiring more training data, often a lot more data.

In fact, some nonlinear algorithms like deep learning methods can continue to improve in accuracy as more data becomes available. So, if a linear algorithm achieves good performance with hundreds of examples per class, one may need thousands of examples per class for a nonlinear algorithm (such as a random forest, or an artificial neural network). This all depends on the dataset itself though and the point made here is that when an (non-parametric) algorithm reaches the plateau, any further training cycles are pure overhead and should be avoided.

3 Modeling Expands the Results of Benchmarks

Preparation and conducting the benchmark test is time consuming. In our case each benchmark test run for 2 hours. Therefore, the benchmark results include the limited number of data points.

In order to expand the results of the benchmark tests we applied ML and Queueing Network Models.

For ML based models the measurement data were split into two data sets: 80% of data were used for model training and 20% of data were used to compare the actual measurement data with prediction results. Trained models are used to predict Response Time, CPU Utilization, I/O rate, Memory Utilization, and Accuracy for each ML Algorithm and ML Library [1].

We evaluated different ML algorithms and selected OLS for prediction of Accuracy and Response Time.

Regression Tree algorithm was selected for predicting CPU and Memory Utilization.

Queueing network models (QNM) were built based on measurement data collected during each benchmark test. Measurement data were aggregated and used as input for QNM to predict the impact of the data size increase on performance and resource consumption of applications using different ML algorithms in Spark environment.

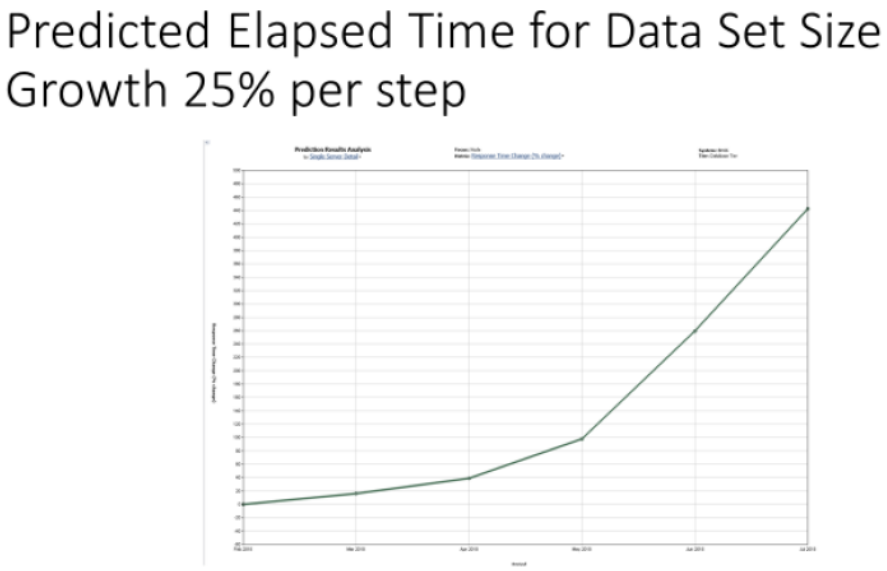

Figure 5. Predicted impact of the data size growth on elapsed time of the application incorporating OLS algorithm

Modeling results expand benchmark results and allow to convert measurement data collected on different infrastructures to the common baseline platform. On another hand same models can be used to convert results of the benchmark tests to the configuration planned to be used for a future ML application. Depending on business requirements applications might have the different priorities to

accuracy or responsiveness or scalability or cost.

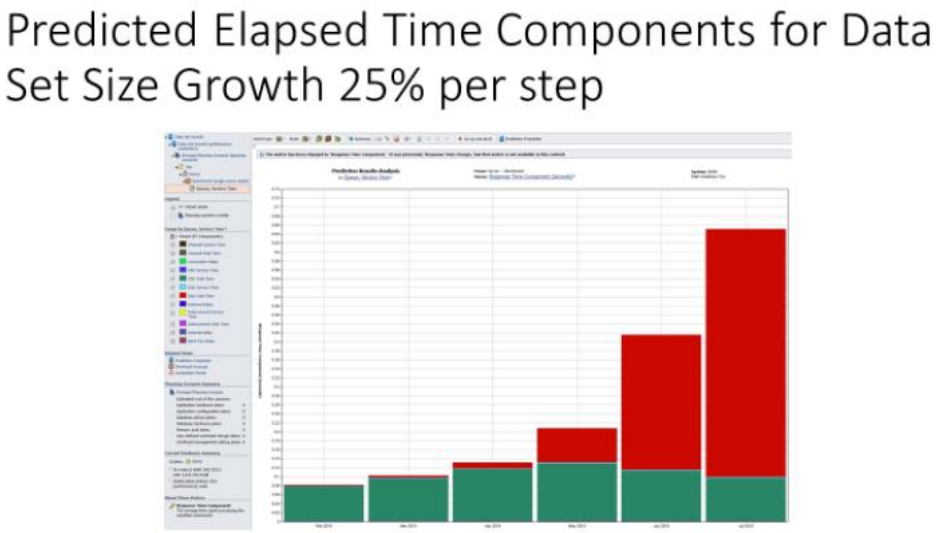

Figure 6. Waiting for I/O will become a bottleneck for application incorporating OLS algorithm with increase of the data set size on current infrastructure

4 Recommender of selection of the appropriate ML algorithm and ML library

Below is an algorithm we implemented to select appropriate algorithm and ML library based on weighted score. The Score takes into consideration the relative importance of the Response time, Accuracy, CPU Utilization, Memory utilization, Number of I/O operations, and other parameters predicted by the model for expected data set size and the number of predictors:

Score = w1 * Accuracy + w2 * Response Time + w3 * CPU Utilization + w4 * Memory Utilization +w5 * Scalability

where: the wi represent relative weight of the corresponding criteria and sum of weighting coefficients *wi* equal to 1.

Response Time can vary between 0 and infinity. We transform the response time as 1 / (1 + RT) to make it as a number between 0 and 1, where 1 is better. In addition to calculating the score we check if predicted CPU Utilization and Memory Usage are less than 1.

Value of score is used to recommend the appropriate ML algorithm and ML Library.

Table 1

ML OLS Algorithm using Python Sklearn ML library is the most appropriate algorithm to satisfy business requirements

Algorithm	library	pred_score	pred_rank	true_score	true_rank
OLS	Python Sklearn	0.962057911	1	0.936165261	1
OLS	Pyspark ML	0.876712666	2	0.753752225	2
Ridge	Python Sklearn	0.781980143	3	0.725268522	3
Ridge	Pyspark ML	0.722426161	4	0.659234146	4
RF	Python Sklearn	0.476284999	5	0.429752013	5
RF	Pyspark ML	0.465422159	6	0.415271967	6

5 Summary

In this presentation we reviewed how data set sizes and number of predictors/attributes affect accuracy, response time, resource utilization and scalability of applications incorporating several types of ML applications.

Models incorporating ML algorithms and QNM were used to expand the benchmark results, convert measurement data collected in different environment to the baseline configuration and estimate infrastructure requirements to meet service level goals for new application after implementation in production environment.

We reviewed an example of the Recommender o select the most appropriate algorithm and library based on requirements to performances, use of resources, and accuracy.

This project is a foundation for organizing collaboration between Universities and Research organizations in benchmarking of ML algorithms and Libraries.

We plan to offer this methodology and Big Data cluster resources for collaborators, including Universities and research organization to organize benchmark testing of other ML algorithms and Libraries in parallel.

Related Work

Selection of the appropriate ML algorithms and libraries during design and development of Big Data application is a part of Performance Engineering, which primary goal is to reduce risk of performance surprises.

Several papers presented at ACM and IEEE conferences discuss Big Data ML algorithms benchmarking, but they are not focus on development recommendation how to select appropriate algorithm and ML library for a specific project.,

Future Work

A future work focus is on benchmarking different types of ML algorithms and incorporation of Prescriptive Analytics to enhance Recommenders

Acknowledgement

Special thanks go out to our colleagues from BEZNext, Leon Katsnelson, CTO, IBM and Leons Petrazickis, Platform Architect, IBM, Protiviti, Dr. Yuri Balasanov, Professor University of Chicago and Dr. Sema Barlas, Director of Master of Science in Analytics program at University of Chicago and Spec Big Data RG.

References

[1]. Boris Zibitsker, BEZNext; Alex Lupersolsky, BEZNext; Dominique Heger, Data Analytica; Yuri Balasanov, University of Chicago; Students of University of Chicago: Jianghui Wen (Cherish); Minghao Bian; Zhiyin Shi, Ziyuan Li. “Selection of Machine Learning Algorithms and Libraries for Big Data Applications. “CMG 2017”

[2]. Boris Zibitsker, Alex Lupersolsky, “Performance Engineering. Modeling Expands Value of Performance Testing for Big Data Applications”

[3]. Daniel A. Menasce “Software, Performance, or Engineering?” WOSP '02 Proceedings of the 3rd international workshop on Software and performance Pages 239-242. B. Zibitsker, IEEE Conference, Delft, Netherlands, March 2016, Big Data Performance Assurance.

[4]. Dominique A. Heger “Big Data Predictive Analytics, Applications, Algorithms and Cluster Systems” ISBN:978-1-61422-951-3.

[5]. Yuri Balasanov, Linear Regression with Large Number of Predictors, Lecture materials for: "Machine Learning and Predictive Analytics", MScA, University of Chicago, 2016

[6]. Benchmark using Reuters Data: https://github.com/BIDData/BIDMach/wiki/Benchmarks#reuters-data.