Term Research Paper – CSP 554
Explore and Evaluate Big Data Machine Learning Tool Kits: Spark MLlib, Mahout, Apache SAMOA, and H2O

- Rakshith Churchagundi Amarnath

Abstract:

Big Data is an extremely large amount of structured and unstructured data, gathered from a wide range of sources increasing by day which often requires fast processing and real-time analysis. In this new context, the performances of the traditional techniques are limited. However, to handle these sizeable quantities of data, new technologies emerged, called Big Data Technologies. One of the important challenges of Big Data is to improve and extend the existing platforms, infrastructures, and standard techniques to manage Big Data, one of them being a Machine Learning platform. There are plenty of tools in Machine Learning that aid in the training of data without being explicitly programmed. Tools are categorized into- framework, platform, library, and interface.

Here, in this paper, I am going to discuss, evaluate, and compare the tools which are for Machine Learning with Big Data that help engineers in building their ML model rapidly. Evaluation along with detailed comparisons of the frameworks are discussed, regarding algorithm availability, scalability, speed, coverage, and more. Some of the major tools highlighting this paper are MLlib, Mahout, H2O, and SAMOA, along with the big data processing engines they utilize. It is hard to find that one tool that "does it all", so this paper provides some insights into each tool's strengths and weaknesses along with guidance on tool choice for certain needs.

Keywords: Machine learning, Big data, Hadoop, Mahout, MLlib, SAMOA, H2O, Flink,

Introduction:

With the advancement of technology in computing, storage, and networking, any high-speed computation on the data available across various data centers is possible today. The amount of data generated by various enterprise applications and social networking is enormous - 40 zettabytes expected by the end of 2020 [5] and expected to grow tremendously in the coming years (175 zettabytes by 2025) [6]. Deriving useful and intelligent information out of this data is of the utmost importance to enhance business value and increase human centricity.

One of the major concerns in the world of Big data is not the collection and storage but evaluation of these data. Machine learning is a top contender amongst the different techniques used to extract knowledge from raw data. In general, the workflow of a supervised machine learning model consists of phases including building the model, evaluating the model, and put the model to production. An example is shown in Fig.1.

Machine Learning algorithms are available, as a subfield of Artificial Intelligence to help derive the intelligence from a plethora of data available across various application domains. Artificial intelligence and Machine learning are trending technologies in the industry to enable business recommendations, predict the future market, etc. The available ML tools have advantages and drawbacks, and many have overlapping uses. Choosing the appropriate tools can be daunting for two reasons. First, the increase in complexity of ML Projects, as well as the data itself, will sometimes require different types of solutions. Second, often developers feel unsatisfied with the tool and some decide to begin one on their own to their needs instead of contributing to the available open-source ones. The goal of this paper is to encourage these decisions by providing a comprehensive review of the current state-of-the-art open-source tools for machine learning.

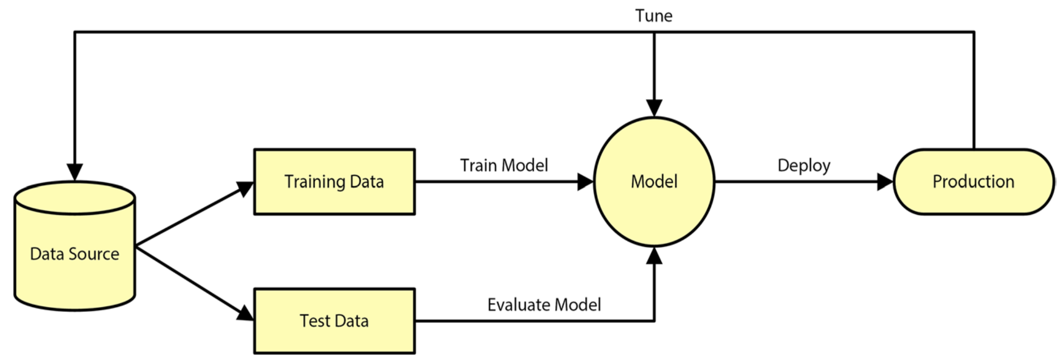

Fig. 1 Supervised Machine Learning Workflow

Section 1 and 2 gave the abstract and introduction to what this paper talks about. The remainder of this paper is as follows; the next section will present a broad understanding of what Big Data is. Section 4 - Machine Learning – An Essential Technique discusses how Machine learning is increasing its popularity day by day and a greater number of companies are applying it with big data. Section Classification of ML tools discusses the different genre of ML tools which are classified based on the feature they have to offer. Section Evaluation Criteria presents the factors which are being considered in this paper to evaluate some of the tools related to ML with big data. Section 7 Machine Learning Tools gives a broad sense about each tool which will be compared in the next section. The following sections are some future work ideas and conclusions of this evaluation which serves as a guide to ML engineers while considering tool for their machine learning needs.

Understanding Big Data:

The term "Big Data" has become a buzzword and often misinterpreted and confused. The frameworks we discuss in this paper are mostly designed with very large data in mind and may not be the best option for certain smaller projects. For this reason, it is very important to choose the right big data frameworks when needed. To do this, it is important to understand big data. This section provides an understanding of what Big Data is and the challenges associated with it.

In 2001, Laney [7] described 3 dimensions of a data management challenge. Commonly referred to as the 3 V's – Volume, Velocity, and Variety can be described as follows:

- The volume – size of the data that is currently in hand requires adequate storage and processing models to develop the tools.
- Velocity – the speed at which data can be processed.
- Variety – structured and unstructured data that is generated by humans and machines.

Since this, numerous people have proposed additions to this list and many refer to four or five V's, adding in Value or Veracity [7 - 8].

The big data problems are Big Data Collections – aggregation of several datasets that are individually manageable, but as a group is too large to fit in one data drive. These data come from different sources, are in contrasting formats, and are stored in separate physical sites and different types of repositories. Today, this problem can be solved by using distributed storage systems.

Big Data Objects - individual datasets that by themselves are too large to be processed by standard algorithms on available hardware. Unlike collections, they typically come from a single source. One solution for big data objects in machine learning is through the parallelization of algorithms. This can be achieved through data parallelism, in which the data is split into more manageable pieces and each subset is computed simultaneously, or task parallelism, where the algorithm is split into steps that can be performed concurrently. It is common nowadays to encounter big collections of big objects as data grows and becomes more widely available. One option is to use tools from the Hadoop Ecosystem.

Machine Learning – An Essential Technique

Machine learning being a subset of Artificial Intelligence is helping us in predicting future actions without much involvement of human beings. So, the goal achievable with Machine learning is that software applications can learn how to make their accuracy better to predict the outcomes

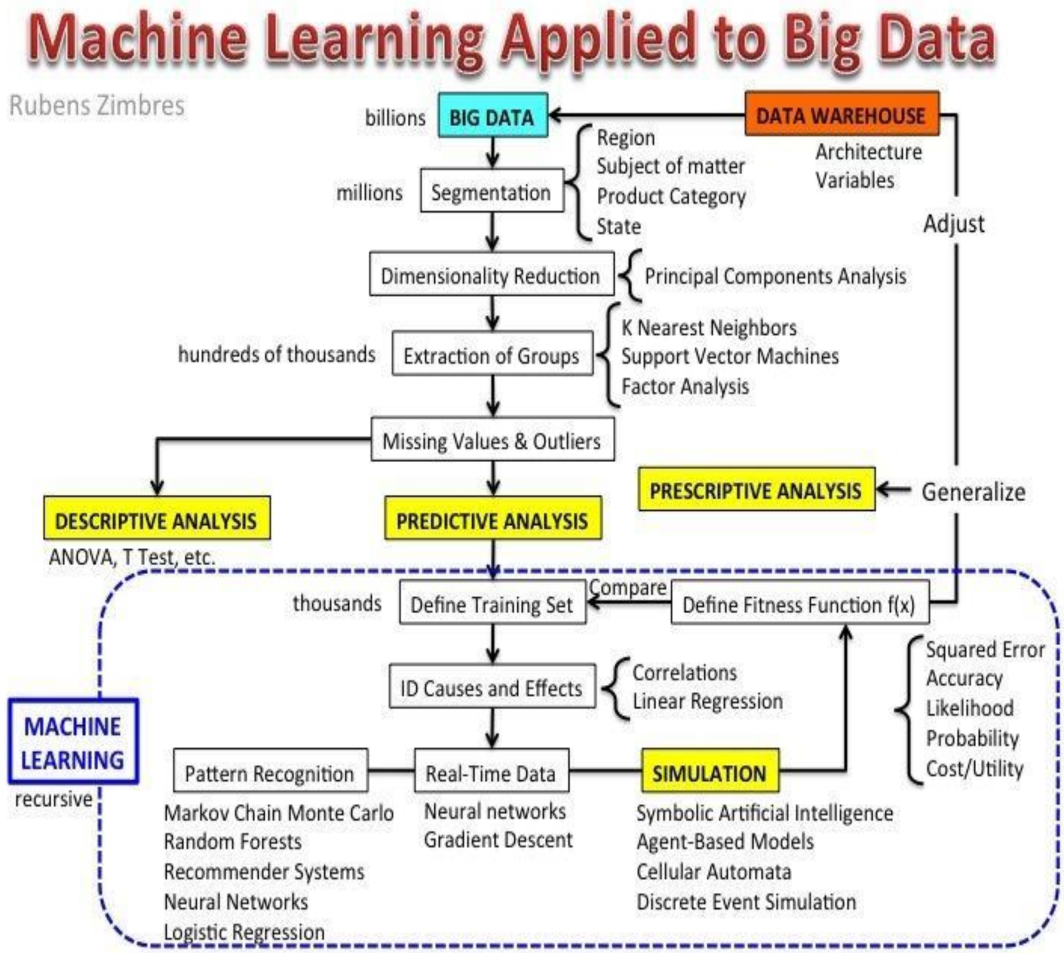

Fig. 2: Machine Learning Applied to Big Data [5]

Big data, the name itself suggests it is basically an analysis of data by discovering hidden patterns or extracting information using tools commonly Hadoop. On the other hand, Machine learning simply teaches computers how to take inputs and gives desirable outputs based on machine learning models.

The usual course of action in big data analytics is all about gathering and transforming the data to extract information. Then the gathered data is used by Machine Learning in order to predict better results. So large enterprises may fulfill their dreams and can get the benefits of big data by properly focusing on machine learning processes but with the help of skilled data scientists in order to run that data into knowledge. In short, it could be said that both big data and machine learning are dependent on each other but still there are some features that make them different.

Although effective, machine learning implementations cannot rely solely on ever-increasing volumes of Big Data and algorithms. The ability to exploit massive quantities of data for machine learning tasks is must-have knowledge for practitioners nowadays.

Data scientist 'Rubens Zimbres' outlines a process for applying machine learning to Big Data in his original diagram in Fig. 2. The most important part is the one where the data scientists need to generate a demand for change in data architecture because this is the part where Big Data projects fail. The orange squares. When algorithms are computationally expensive or when infrastructure is not ready for ML algorithms [5].

Zimbres' process includes paths for descriptive, predictive, and prescriptive analysis, as well as simulation. Importantly, the machine learning process is explicitly noted as recursive, which holds true in modeling large quantities of data. While Zimbres himself states that there are a few small blunders with the process in the graphic, all in all, it represents a relevant high-level roadmap. It should be useful for newcomers to the data science field.

The core notion of the whole ML concept is the model's capacity to automatically apply sophisticated mathematical computations to large data repeatedly before a most possible answer is found

Machine Learning tool Kits:

Machine Learning tools can be a platform, library, an interface, or any local or remote tool. These toolkits help the developers in delivering Machine Learning models rapidly and efficiently without stepping into the details of the core algorithms involved, hence providing a well-defined and brief approach for classifying ML models by applying a set of pre-built and improved modules [2].

The tools can be divided into four branches, as depicted in Figure 3 below.

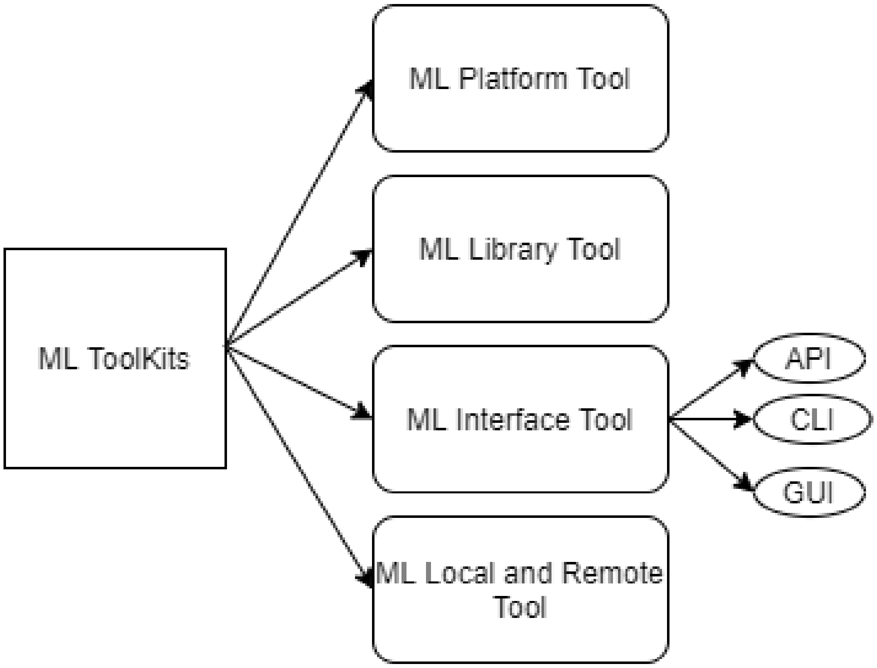

Fig. 3 – Classification of ML Toolkits'

ML Platform:

A platform for automating and accelerating the delivery lifecycle of predictive applications capable of processing big data using machine learning or related techniques**.** [8]. ML platforms provide complete facilities needed at every step of an ML project development. An interface may include API, CLI or GUI, or a combination of these while programming. They are used for general-purpose modeling, instead of focusing on accuracy, scalability, and speed. Developers can congregate all the loose components or features as per their specific needs.

ML Library

Libraries are sets of routines and functions that are written in each language. A robust set of libraries can make it easier for developers to perform complex tasks without rewriting many lines of code. Machine learning is largely based upon mathematics.

ML library interface is usually an API that involves programming. They are designed keeping in mind a use-case achieve specific results.

ML Interfaces:

These can be branched into three main parts namely, APIs, CLIs, and GUIs

i. ML API (Application Programming Interface):

ML tools support an API that provides the ability to decide what components to work with and how to apply them in the ML programs. It provides the capability of developing our own ML tools. Its main used can be used to build our processes, and thereby, it can be further implemented on ML projects to automate them in an improved way. It gives the developers their flex in developing their own methods, customizing them with existing libraries and methods.

ii. ML CLI (Command Line Interface):

It focuses on input and output i.e. it structures ML tasks in terms of the required input and output to be produced [2]. Besides, it also comprises of command line parameterization and command-line programs. Even non-programmers can benefit this to perform their tasks through ML projects. It stores the commands and command-line arguments. It is helpful in creating small subtasks of ML project.

iii. ML GUI (Graphical User Interface):

Machine learning tools support a GUI which mainly focuses on the graphical representation of data i.e. visualization and consists of windows, point, and click [2].

The new programmers can really benefit from ML GUIs as it gives an environment where they can complete their tasks easily through ML. If used right, we can extract maximum information.

ML Local and Remote Tools:

Machine learning Local tools can be downloaded, installed, and run in our local environments and machines. The process can be pretty straight forward. It utilizes our machine's own memory.

It gives us control over the parameters to devise predictions on newer data thereby supporting the run configuration of the system.

Machine learning Remote runs on a server of a third-party. We can perform actions by calling it to our local environment through RPCs (Remote Procedure Calls). These typically known as Machine Learning as a Service (MLaaS) [2]. Remote tools can handle large datasets even though the data scales up rapidly. It helps when our local machines are not as powerful as it needed to run these programs quickly by providing quite a few cores. It also helps in running amongst multiple machines. One downfall for these tools might be the smaller number of ML algorithms it supports since complex modifications are needed.

Although there are quite a few ML tools are available in various platforms to us, most often developers reject them as they lack some needed features or are difficult to integrate into an existing environment. Also, many people might lack a full understanding of the capabilities of these various platforms. This is heightened by the fact that there has been little comprehensive research into many popular frameworks. Developers might move to a different tool before the evaluation of tools comes out published as extensive researchers need time to evaluate it thoroughly by overviewing different aspects of the features the tool provides.

Evaluation Criteria for ML Toolkits:

The selection of a specific ML Toolkit for our Big Data comes down to several factors which are dependent mainly on the project itself. Some of the factors which are used in this paper are as follows:

Scalability:

It is defined as the capacity to be changed in size or scale. So, in our scenario, this should be with respect to both the size and complexity of data – both now and how data might be soon. This becomes one of the important factors because some ML toolkits might be best built for big data and might not be appropriate for small data and perform poorly. This condition holds true for data dimensions also.

Speed:

When it comes to speed in the Machine Learning world, there can be two factors to consider while choosing an ML tool for any job. One being the Execution Speed – time frame with which the algorithm executes and classifies via trained model.

The other is the Training Speed or Learning Rate – This factor is more emphasized than the previous as this. It is a tuning parameter in an optimization algorithm which heavily depends on the kind of the problem to be solved, the trained data, and the images. This factor plays a major role in models that are updated frequently, not so much for batch processing systems.

Usability:

This is simply how the ML tool is of convenience to the user to set up and run, programming languages available, the quality of documentation it provides, whether it has a UI ready for non-programmers. One of the major factors to weigh in this is the amount of support it gets. Sometimes a huge availability of a knowledgeable user community forums helps in numerous occasions as it involves developers own previous experiences.

Scope:

It is always a good thing when our ML tool offers a wide range of options related to the different ML classes and a variety of implementation options in each class. Some of the non-distributed frameworks, Weka for one provides diverse selections to the user but their scope is limited to a few algorithms [9]. Many of the tools are difficult to set up and learn, it is important to consider future needs as well as current.

Flexibility:

Machine learning tasks do not fit in all instances ranging from as simple as setting the value of *k* in a k-means clustering task or building a troupe of learners [9], most jobs will require some amount of customization in terms of parameter tuning before a model is deployed. Also, how well the tool behaves in a new environment or a system.

Other factors we should weigh in while considering ML tool for out big data are: Cost [7] – while the majority of them are open source, there are some which provide a premium subscription which adds a support layer for the user.

Machine Learning Tools:

This section provides an in-depth look at the strengths and weaknesses of the various machine learning tools available. For a complete look at how the tools and engines fit together, Fig. 4 shows an outlook of how the tools and engines fit together and provide the resources which are common between multiple tools like the Implementation Algorithms. It also shows the direct dependent engines needed to run and the compatible engines for some ML Tools. and the algorithms they implement.

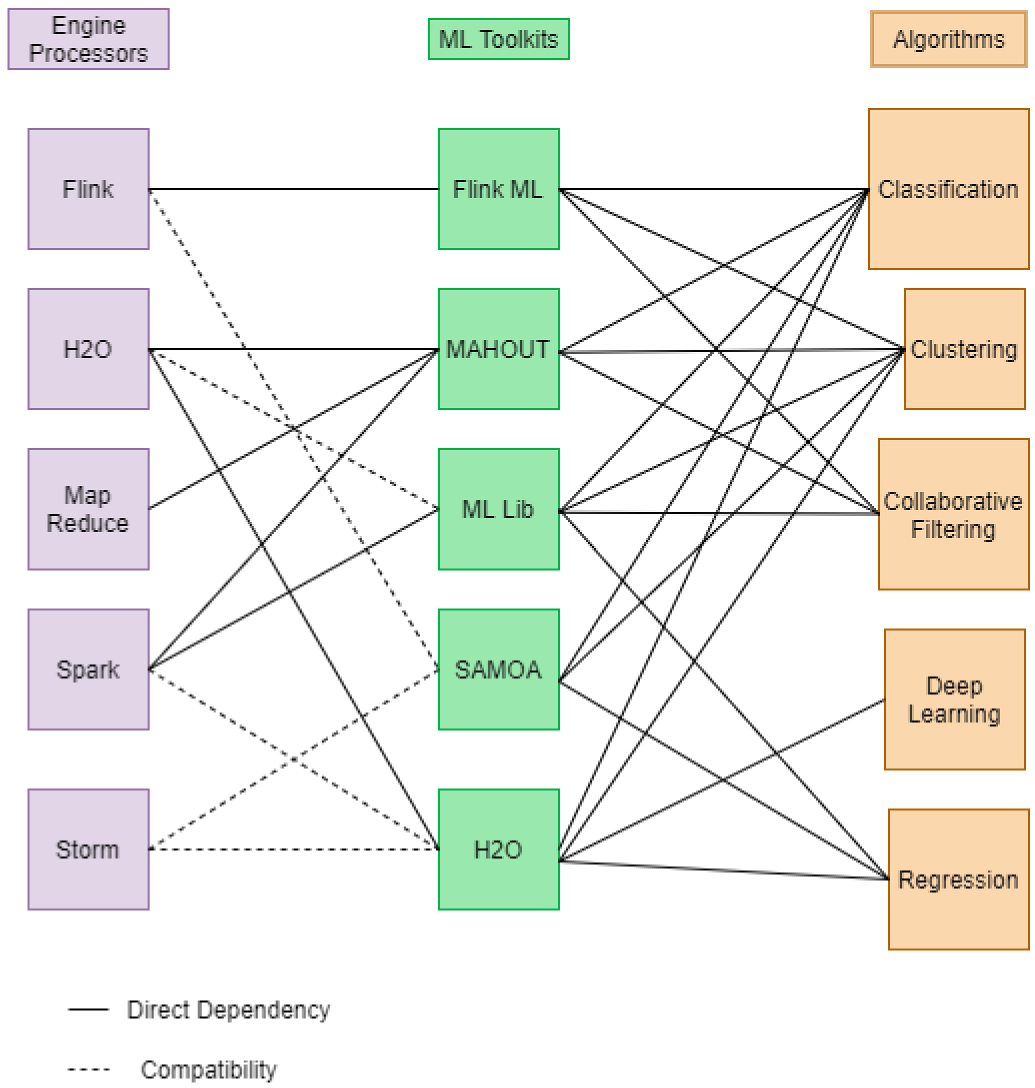

Fig. 4 Machine Learning tools with their engine processor and core algorithm

MLlib:

MLlib is Spark's scalable machine learning library consisting of common learning algorithms including classification, regression, clustering, collaborative filtering, Additional tools include dimensionality reduction, feature extraction and transformation, optimization, and basic statistics [10]. MLlib is considerably a young tool, their official notes say MLlib is under active development. The APIs provided by Spark for developers/experimentalists may change in future releases. It is significantly faster than MapReduce [1]. They also provide good documentation for all the changes it had in between releases. A good point to always consider is that you may face some issues when performing Machine Learning on multiple platforms as MLlib is tied to Spark.

Setting up Spark MLlib is relatively easier as the larger part of the library we need to set up comes in-built with its processing engine thus steering clear of configuration issues that one might face with Mahout. The amount of optimization it provides also helps in writing new implementations and customizing some existing algorithms, although unsure of the actual number increase in efficiency it provides in the latter if the data has higher dimensions. The documentation as said earlier comes with adequate information about different releases, the user community will eventually grow as more developers migrate to Spark from MapReduce. The notable companies that use Spark MLlib for their recommendation engines are OpenTable and Spotify [11].

Mahout:

Mahout covers the same range of algorithms as MLlib. It is a very rich and complete library of Machine Learning which covers the algorithms of classification and clustering. It distributes its computational algorithms using the Map/Reduce paradigm [1]. When you get to a point in your code where you are ready to math it up, you can use its math environment called Samsara [3] where you can elegantly express yourself Mathematically. It includes linear algebra, statistical operations, and data structures. They provide distributed algorithms to users, rather than simply a library of already-written implementations.

Mahout works well with MapReduce and spark as the algorithms are optimized to work that way. Integrations with H2O and Flink are also developed and available [3]. One of the common rants in the user community is that it is difficult to set up on an existing Hadoop cluster [9]. The updates regarding documentation are archaic and it's less relevant to the current versions, however, there will be an update eventually. The number of active committers is very low, with only a handful of developers making regular commits [9]. The timing between production versions takes longer than other tools.

The algorithms included in Mahout focus primarily on classification, clustering, and collaborative filtering, and have been shown to scale well as the size of the data increases. One of Mahout's most cited resources is its great outcomes in building off the standard baseline algorithms. In this case, to take any advantage of this adaptability, solid capability in Java programming is required. A few analysts have shown trouble with an arrangement or with merging it into an existing environment. On the other hand, several companies have reported success using and customizing Mahout into production. Notable examples include Mendeley, LinkedIn, and Overstock.com [9], who all use its recommendation tools as part of their big data ecosystems.

SAMOA:

Apache SAMOA (Scalable Advanced Massive Online Analysis) is an open-source platform written in Java originally developed at Yahoo! Labs in Barcelona in 2013 [9]. It can be used for mining big data streams. SAMOA provides a few algorithms such as classification, clustering, and regression to tackle data mining and machine learning problems. It also provides programming abstractions to develop new algorithms for developers. Its features include a pluggable architecture that allows it to run on several distributed stream processing engines such as Apache Flink, Storm, and Samza. It does not yet have a large active community, yet it offers detailed documentation.

This platform might be a good fit for you If your data is being updated in real-time. SAMOA guarantees flexibility to users. The developer API which it provides allows the user to implement new algorithms and can be made to run on different processing engines. A researcher implemented a modeling algorithm and compared experimental results using both SAMOA and MOA (Massive Online Analysis). Results indicated significantly higher throughput on SAMOA and showed the framework to be robust and stable [9].

Even though it offers fewer ML tools, they cover many of the common ML tasks like clustering, classification using CluStream, and decision trees [12]. It also offers a plugin called SAMOA-MOA [12] which allows the use of MOA classifiers and clustering algorithms inside the SAMOA platform, but it will not be distributed because of MOA's algorithms which are not distributed. Some of the projects have leveraged SAMOA as part of frameworks for efficiently finding top-k items and for recognition of internet traffic patterns [9].

H2O:

H2O is an open-source, in-memory, distributed, fast, and scalable predictive analytics and Machine Learning Platform [1]. H2O can be considered as a product, rather than a project. H2O.ai also offers an enterprise edition with 2 tiers of support, however, most of their offering features can be found in their open source for free. Additionally, it offers a web-based user interface, making learning tasks more accessible. Analysts and Statisticians who may not have strong programming backgrounds can certainly use his to their advantage. For all the coders out there, it also supports Java, R, Python, and Scala. The machine learning tools offered to cover a range of tasks, including classification, clustering, generalized linear models, statistical analysis, ensembles, optimization tools, data preprocessing options, recommendation, time-series, and deep neural networks.

They offer seamless integration with R and R Studio, as well as Sparkling Water for integration with Spark and MLlib[9]. The documentation is thorough, and their support is quick in answering questions and feedback. More practical results are needed to be compared and evaluated with respect to speed, performance, and reliability. One notable example of a company using H2O in production is ShareThis [9], which uses modeling for prediction

Comparison of Machine Learning Toolkits:

According to the evaluation criteria's chosen in the earlier section of this paper, we can come to a rating system for each tool based on the already accessible contents. A comparison of the tools concerning Scalability, speed, scope, usability, and flexibility is shown in Fig. 5. The graph is meant to be viewed as where each tool stands with each other with respect to the selection criteria that have been outlined in this paper.

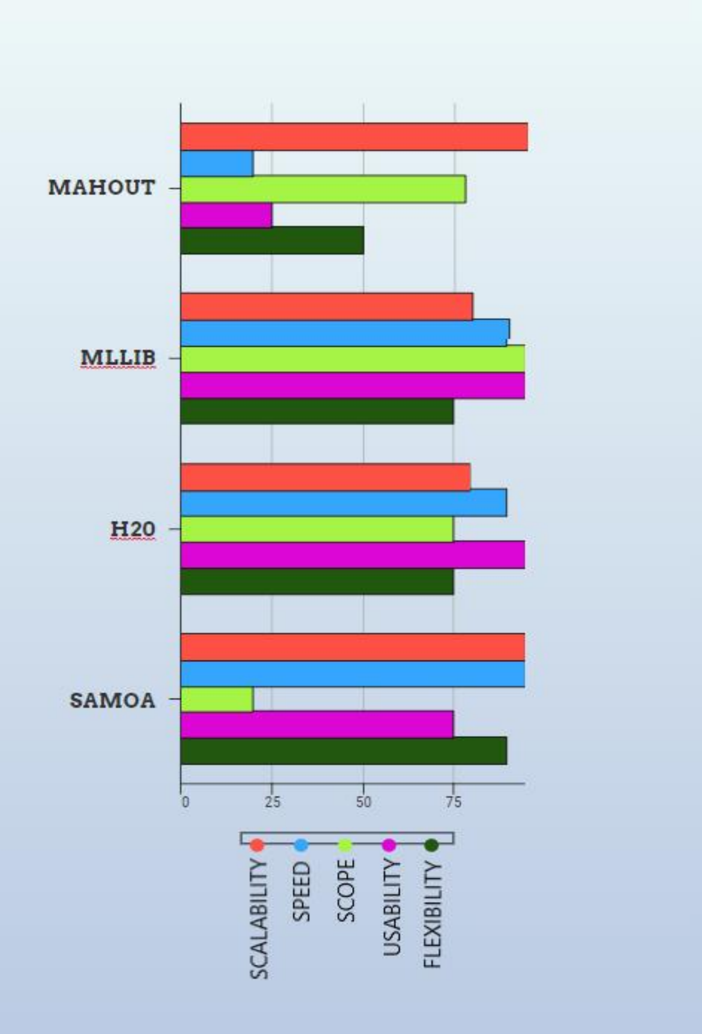

Fig. 5 Comparison of ML Toolkits

The choice of ML tool will depend on each user's requirement. From the graph and the insights at hand, MLlib and H2O are very good options for general needs. These fast tools which also scales very well and works with different dataset sizes are MLlib and H20. These also have a rich selection of algorithms. MLlib works for the most part and it fulfills the basic ML tasks you need to perform anytime. However, H2O has its upper hand when it comes to deep learning. Both tools have APIs for programming in multiple languages, and H2O also offers a GUI, making it effortless to use for those without coding proficiency.

If your needs require you to write and execute your implementation algorithms, then SAMOA and Mahout is the way. They offer future flexibility if you need at later stages in your project. These factors weigh is when your data and needs are always changing. SAMOA help in creating streaming algorithms that are created keeping real-time in mind, whereas Mahout offers batch implementations. SAMOA is the fastest and most scalable option amongst these tools.

Additional Tools:

The Machine Learning tools discussed in the above have been chosen due to their widespread use or versatility with respect to implemented tools on a range of applications. This is by no means a comprehensive list of all open-source learning tools for Big Data and there are additional frameworks that show promise for large-scale machine learning tasks as well.

For example, “Flink-ML” – an ML library for Flink. It is a new effort in the Flink community, with a growing list of algorithms and contributors. With FlinkML we aim to provide scalable ML algorithms, an intuitive API, and tools that help minimize glue code in end-to-end ML systems [4]. It supports implementations of Logistic Regression, k-Means Clustering, and Alternating Least Squares (ALS) for the recommendation. It also supports Mahout’s DSL (Domain Specific Language) for linear algebra which can be used for the optimization of learning algorithms.

Future work:

Future work will include quantitative comparisons of these tools based on formally defined criteria, and also experimental results on a common ground of data, but for this survey, the qualitative rankings is based on the exposure to each tool and related works. While several other tools are available, some of which are discussed, for other new tools, there is not yet enough literature to properly evaluate them.

CONCLUSION:

The amount of data surge within the next few years will be enormous as we all are aware just by looking at the current numbers and growth. Being prepared with a good base of knowledge of which tools to use could probably save a great amount of time for Data Scientists. Big Data technologies through its ecosystem are available to aid us during these challenges. This paper presented the different Machine Learning toolkits associated with big data and also its evaluation by looking it into pros and cons, giving which tool has the upper hand when it comes to a certain factor thus proposing which tool to use for specific requirements.

If the performance of traditional Machine Learning algorithms is not efficient enough, we can always choose the tool which gives us the option of Flexibility to make them more suitable for ever-changing data needs. Data architects have very challenging work also in terms of the time they must put the right architecture in place when they have limited options to play with and experiment before production. As machine learning principles are gradually being applied in active research and production Environments it is becoming necessary to provide resources to promote learning activities to ensure the right fit in the later stages. Many of these tools are still improving in time and further research work is required to better benchmark and analyze the different experiments.

The paper discusses how Machine learning is an essential technique that can be applied to Big Data, then the classification of ML tools. Then how some tools are restricted to one processing engine and others are compatible with multiple engines, the algorithms each tool comes with.

Additionally, a list of criteria for evaluation and selection of machine learning toolkits with a discussion of their advantages and drawbacks. For example, Mahout and MLlib have the options for recommendations, so if the intended application is an e-commerce site or a streaming service, one can choose from them for its recommendation features. That said recommendation uses real-time data, so engineers must think ahead and calculate all the future possibilities. H2O comes with a built-in GUI and holds up with deep learning which makes H2O the only tool which has both. The most versatile tools must be Mahout and MLlib as it has well-rounded libraries, algorithms, and has the compatibility to work well with Spark and H2O. MLlib has a broad selection of algorithms and there is a lot of manpower invested in this tool to make it more well-defined. Mahout and MLlib are math friendly tools where users can create their own algorithms. Real-time learning is rising in importance and might increase its compatibility with a greater number of tools. In general, the usage of these tools is also on the rise. In time and we can hope to see these tools being developed more and checking all the boxes.

References:

[1] Sassi, Imad & Anter, Samir. (2019). A STUDY ON BIG DATA FRAMEWORKS AND MACHINE LEARNING TOOLKITS. 61-68. 10.33965/bigdaci2019_201907L008.

[2] Khan, Afreen & Zubair, Swaleha. (2018). Machine Learning Tools and Toolkits in the Exploration of Big Data. International Journal of Computer Sciences and Engineering. 6. 570-575. 10.26438/ijcse/v6i12.570575.

[3] Mahout http://mahout.apache.org/

[4] ci. Apache.org https://ci.apache.org/projects/flink/flink-docs-release-1.4/dev/libs/ml/

[5] https://www.kdnuggets.com/

[6] https://www.networkworld.com/article/3325397/idc-expect-175-zettabytes-of-data-worldwide-by-2025.html

[7] https://www.softwaretestinghelp.com/machine-learning-tools/

[8] https://www.cloudpulsestrat.com/posts/machine-learning-platforms-predictive-applications-part-2-machine-learning-platform

[9] https://www.semanticscholar.org/paper/A-survey-of-open-source-tools-for-machine-learning   Landset-Khoshgoftaar/19469939ee1c6cb51db757b71338c61fcfe8ee76

[10] https://spark.apache.org/docs/1.1.0/mllib-guide.html

[11] https://cwiki.apache.org/confluence/#all-updates

[12] https://samoa.incubator.apache.org/documentation/