BIG DATA MINING TOOLS

Prathmesh Kulkarni

Master of Computer Application
Sardar Patel Institute of Technology
Mumbai, India

***Abstract*—Big data is being developed by all that is around us all time. Every digital process and social media transfer generates it. Systems, sensors and mobile devices carries it. Big data is turning up from many sources at an alarming velocity, volume and variety. To extract meaningful value from big data, you need optimal processing power, analytics capabilities and skills. This paper compares big data mining tools, their working and under what environment which tool should provide maximum output. I have taken three tools for comparison i.e. Hadoop, Weka, RapidMiner(Raddop), Apache Spark.**

Keywords—Big Data; mining; comparison (key words)

I. INTRODUCTION (*HEADING 1*)

It is legitimately said that data is money in today's world. Along with the changeover to an app-based world comes the epidemic advancement of data. However, most of the data is unstructured and hence it takes a process and method to extract sensible information from the data and convert it into simple and usable form. This is where data mining comes into picture. Ample of tools are available for data mining tasks using artificial intelligence, machine learning and other techniques to extract data. In this paper I have tried to compare working of few tools build upon upon size of data, requirement from user side etc.

II. BIG DATA

A. *Definition*

Massive volume of both structured and unstructured date that is so huge that it is difficult to process using traditional database and software techniques.

B. *Types of Big Data*

There are two types of big data: structured and unstructured.
Structured data are numbers and words that can be easily classified and analyzed. These data are developed by things like network sensors installed in electronic devices, Smartphone's, and global positioning system (GPS) devices. Structured data also incorporate things like sales figures, account balances, and transaction data. [1]
Unstructured data incorporate more complicated information, such as customer analysis from commercial websites, photos and other multimedia, and comments on social networking sites. These data cannot easily be separated into classes or analyzed numerically.[1]

III. HADOOP

In the Big data world the sheer volume, velocity and variety of data renders most ordinary technologies ineffective. Thus in order to overcome their helplessness companies like Google and Yahoo! needed to find solutions to manage all the data that their servers were gathering in an efficient, cost effective way.

A. Introduction to Hadoop

Hadoop was basically developed by a Yahoo! Engineer, Doug Cutting as a counter-weight to Google's BigTable. Hadoop was Yahoo!'s pursuit to break down the big data problem into small pieces that could be handled in parallel. Hadoop is now an open source project accessible under Apache License 2.0 and is now widely used to manage large chunks of data successfully by many companies.

B. Use of Hadoop

- Searching - Yahoo, Amazon, Zvents
- Log Processing - Facebook, Yahoo, ContextWeb. Joost, Last.fm
- Data Warehouse - Facebook, AOL
- Video and Image Analysis - New York Times, Eyealike

C. Pillars of Hadoop

At its crux, Hadoop has two prime systems:
Hadoop Distributed File System (HDFS): the storage system for Hadoop spread out over numerous machines as a means to trim cost and increase authenticity.
MapReduce engine: the algorithm that filters, sorts and then uses the database input in some way.

D. *HDFS Working*

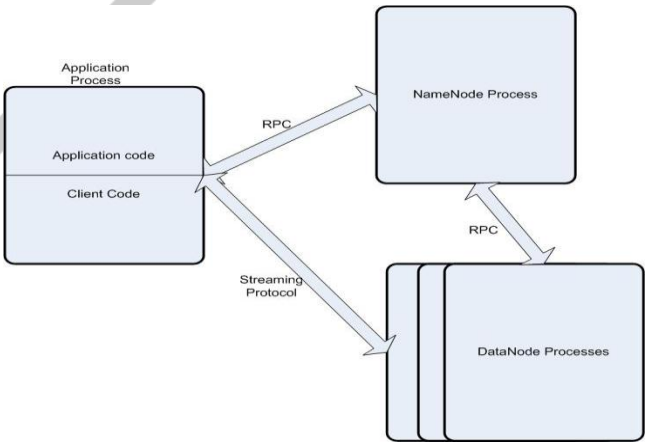

Figure 1. HDFS Working

With the Hadoop Distributed File system the data is written once on the server and afterwards read and re-used many times following. When contrasted with the repeated read/write actions of most other file systems it explains part of the speed with which Hadoop operates. As we will see, this is why

HDFS is an admirable choice to deal with the high volumes and velocity of data required.

The way HDFS works is by having a main NameNode and multiple data nodes on a commodity hardware cluster. All the nodes are usually organized within the same physical rack in the data center. Data is then broken down into separate blocks that are distributed among the various data nodes for storage. Blocks are also replicated across nodes to reduce the likelihood of failure.

The NameNode is the smart node in the cluster. It knows exactly which data node contains which blocks and where the data nodes are located within the machine cluster. The NameNode also manages access to the files, including reads, writes, creates, deletes and replication of data blocks across different data nodes.

The NameNode operates in a "loosely coupled" way with the data nodes. This means the elements of the cluster can dynamically adapt to the real-time demand of server capacity by adding or subtracting nodes as the system sees fit.

The data nodes constantly broadcast with the NameNode to see if they need complete a certain task. The unbroken communication makes sure that the NameNode is aware of each data node's status at all times. Since the NameNode appoints tasks to the individual datanodes, should it realize that a datanode is not working properly it is able to immediately re-assign that node's task to a different node containing that same data block. Data nodes also communicate with each other so they can collaborate during normal file operations. Clearly the NameNode is demanding to the entire system and should be duplicated to avoid system failure.

Again, data blocks are replicated across multiple data nodes and access is managed by the NameNode. This means when a data node no longer sends a "life signal" to the NameNode, the NameNode unmaps the data note from the cluster and keeps operating with the other data nodes as if nothing had happened. When this data node comes back to life or a different (new) data node is detected, that new data node is (re-)added to the system. That is what makes HDFS resilient and self-healing. Since data blocks are replicated across several data nodes, the failure of one server will not corrupt a file. The degree of replication and the number of data nodes are adjusted when the cluster is implemented and they can be dynamically adjusted while the cluster is operating.

Data integrity is also anxiously controlled by HDFS's many capabilities. HDFS uses transaction logs and validations to make sure integrity across the cluster. Usually there is one NameNode and possibly a data node running on a physical server in the rack, while all other servers run data nodes only.[7]

E. Advantages of Hadoop

1.Scalable

Hadoop is a highly scalable storage platform, because it can stock and assign very large data sets across hundreds of cheap servers that work in parallel. Unlike conventional relational database systems (RDBMS) that can't scale to process large amounts of data, Hadoop enables businesses to run applications on thousands of nodes involving thousands of terabytes of data.[2]

2. Cost effective

Hadoop also offers a cost effective storage solution for businesses exploding data sets. The problem with traditional relational database management systems is that it is extremely cost prohibitive to scale to such a degree in order to process such massive volumes of data. In an effort to reduce costs, many companies in the past would have had to down-sample data and classify it based on certain assumptions as to which data was the most valuable. The raw data would be deleted, as it would be too cost-prohibitive to keep. While this approach may have worked in the short term, this meant that when business priorities changed, the complete raw data set was not available, as it was too expensive to store. Hadoop, on the other hand, is designed as a scale-out architecture that can affordably store all of a company's data for later use. [2]

3.Flexible

Hadoop enables businesses to comfortably access new data sources and tap into different types of data (both structured and unstructured) to develop value from that data. It means businesses can use Hadoop to derive important business insights from data sources such as social media, email conversations or clickstream data. [2]

4.Fast

Hadoop's exclusive storage method is based on a distributed file system that basically 'maps' data wherever it is located on a cluster. The tools for data processing are often on the same servers where the data is located, resulting in much faster data processing. If you are dealing with large volumes of unstructured data, Hadoop is able to efficiently process terabytes of data in just minutes, and petabytes in hours.[2]

5. Resilient to failure

A key asset of using Hadoop is its fault tolerance. When data is sent to an individual node, that data is also replicated to other nodes in the cluster, which means that in the event of failure, there is another copy available for use.[2]

F. Disadvantages of Hadoop

As the heart of so many implementations, Hadoop is almost synomous with big data.

1. Security Concerns

Just administrating a complex applications such as Hadoop can be demanding. A simple example can be seen in the Hadoop security model, which is disabled by default due to sheer complexity. If whoever administrating the platform lacks of know how to enable it, your data could be at great risk. Hadoop is also missing encryption at the storage and network levels, which is a major selling point for government agencies and others that prefer to keep their data under wraps.

2. Vulnerable By Nature

Speaking of security, the very makeup of Hadoop makes running it a risky proposition. The framework is written almost entirely in Java, one of the most widely used yet controversial programming languages in existence. Java has been heavily exploited by cybercriminals and as a result, implicated in many security breaches.

3. Not Fit for Small Data

While big data is not exclusively made for huge businesses, not all big data platforms are suited for small data needs. Unfortunately, Hadoop happens to be one of them. Due to its high capacity design, the Hadoop Distributed File System, lacks the ability to efficiently support the random reading of small files. As a result, it is not recommended for organizations with small quantities of data.

4. Potential Stability Issues

Like all open source software, Hadoop has had its fair share of stability issues. To avoid these issues, organizations are strongly recommended to make sure they are running the latest stable version, or run it under a third-party vendor furnished to arm such problems.

G. When to use Hadoop

1.Data sets are colossal in size

2. You have fascinating programming skills

3. If you want to build Enterprise Data Hub for the future

4. Historical is data is as important as current data

IV. WEKA

A. Introduction

Waikato Environment for Knowledge Analysis (Weka) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. It is free software licensed under the GNU General Public License.[4]

B. Advantages of Weka

1. Freely available under the GNU General Public License.

2. Portability: Since it is implemented in the Java and thus runs on almost any computing platform.

3. A comprehensive collection of data pre-processing and modeling techniques.

4. Ease of use due to its GUI (graphical user interfaces).

C. Working

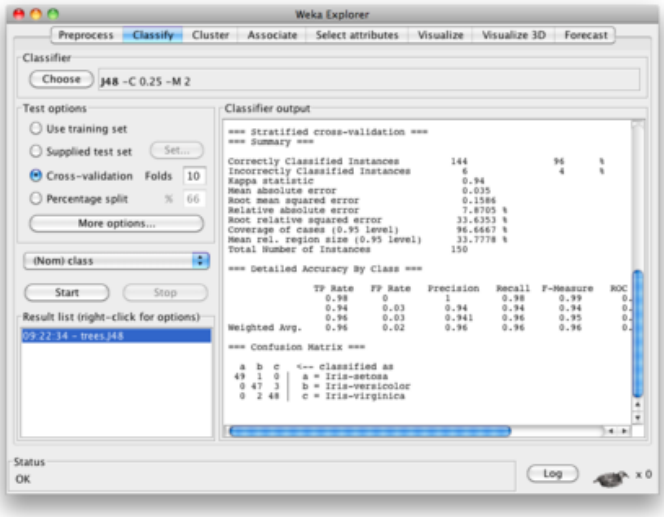

Figure 2. Interface of Weka Tool

WEKA is java based open source data mining tool which has collection of data mining algorithms such as lazy, rules, decision trees and so on. WEKA opens with 4 options (Explorer, Experimenter, KnowledgeFlow and Simple CLI). Mainly, Explorer and Experimenter are used for data mining. For multiple algorithms comparison, Experimenter is used but for specific results of data mining, Explorer is used. Explorer opens with a screen of data preprocessing. The difficulty on WEKA is opening file because most of data sets are in excel and excel can turn into CSV format but excel file is semicolon but CSV must be for comma, so it needs to convert in text file format but it takes time. However; information on algorithms (capabilities and descriptions) and user options are the best features about WEKA, especially any user can use without any training as well as user can implement its own algorithm. [8]

V. RAPIDMINER

A. Introduction

It is written in the Java Programming language, this tool offers advanced analytics via template-based frameworks. Users hardly have to write any code. Offered as a service, instead of a piece of local software, this tool holds top position on the list of data mining tools.[6][4]
RapidMiner Radoop provides an easy-to-use graphical interface for analyzing data on a Hadoop cluster. It desires a accurately configured Hadoop cluster with a running Hive server.

B. RapidMiner Functions

In addition to data mining, RapidMiner also provides functionality like

1. Data pre-processing: Data pre-processing portray any type of processing performed on raw data to amass it for another processing procedure. Often used as a preliminary data mining practice, data pre-processing translates the data into a format that will be more easily and effectively processed for the user needs.

2. Data visualization: Data visualization is a common term that depicts any effort to help people figure out the connotation of data by putting it in a visual context. Patterns, trends and correlations that might go undetected in text-based data can be defined and noticed easier with data visualization software.

3. Predictive analytic s: Predictive analytic s is the practice of obtaining information from existing data sets in order to determine patterns and predict future outcomes and trends. Predictive analytics does not tell you what will happen in the future. It forecasts what might happen in the future with an certain level of reliability, and includes what-if scenarios and risk assessment.

What makes it even more commanding is that it add learning schemes, models and algorithms from WEKA and R scripts.
RapidMiner is distributed under the AGPL open source licence and can be downloaded from SourceForge where it is rated the number one business analytics software.

B. RapidMiner Radoop

1. Big Data Predictive Analytics

RapidMiner is at the forefront of Hadoop and Spark predictive analytics, bringing ease-of-use and visual analytics design to data scientists and analysts who are looking to extract value from their Big Data. RapidMiner Radoop, a core component of the RapidMiner Predictive Analytics Platform, extends predictive analytics to Hadoop and strongly supports Hadoop security implementations, all while delivering a seamlessly experience.
RapidMiner Radoop converts the predictive analytics workflows you design in RapidMiner Studio into the language of Hadoop. Radoop speaks native Hive, MapReduce, Spark, Pig and Mahout, ensuring that each step in the predictive analytics process is correctly integrated and executed across core Big Data technologies. This lets you focus on developing competitive analytics, rather than on programming Hadoop.

2. Insights into ALL your data

RapidMiner's Radoop naturally creates and executes an optimal analytics plan for your Hadoop. It pushes analytic instructions into Hadoop and Spark where predictive analytics are executed across the entire cluster, taking advantage of the processing power of Hadoop. This enables analysis upon the full breadth and variety your Big Data, as compared to solutions that can only analyze subsets and pieces of extracted Hadoop data.

3.Supports R and Python Scripts

Radoop provides you added flexibility, letting you absorb your favorite SparkR, PySpark, Pig and HiveQL scripts within your predictive analytics workflows. By adding SparkR and PySpark support in Radoop 2.6, RapidMiner becomes the first and only visual predictive analytics solution to combine data preparation on Spark, predictive analytics using Spark's Machine Learning library (MLlib), and the ability to incorporate custom-built R and Python scripts.

Extract the full value from your Big Data in Hadoop with powerful predictive analytics while realizing up to a 20X performance increase over traditional Hadoop approaches.

For "Kerberized" Hadoop clusters, RapidMiner Radoop integrates with Kerberos authentication so that users and their workflows can access necessary Hadoop services. RapidMiner Radoop also supports data access authorization employing Apache Sentry and Apache Ranger. All administration and configuration are reduced to a minimum for IT, while complexity is handled behind the scenes so users only see necessary settings.[12]

V. APACHE SPARK

A. Introduction

Apache Spark started as a research project at UC Berkeley in the AMPLab, was started with a goal to design a programming model that supports a much wider class of applications than MapReduce, while maintaining its automatic fault tolerance. Spark offers an abstraction called Resilient distributed Datasets (RDDs) to support these applications efficiently. RDDs can be stored in memory between queries without requiring replication. Instead, they rebuild lost data on failure using lineage: each RDD remembers how it was built from other datasets (by transformations like map, join or groupBy) to rebuild itself. RDDs allow Spark to outperform existing models by up to 100x in multi-pass analytics. RDDs can support a wide variety of iterative algorithms, as well as interactive data mining and a highly efficient SQL engine Shark.

B. Difference between Hadoop and Spark

1. Performs different tasks
Hadoop and Apache Spark are both big-data frameworks, but they don't really serve the same purposes. Hadoop is essentially a distributed data infrastructure: It distributes massive data collections across multiple nodes within a cluster of commodity servers, which means you don't need to buy and maintain expensive custom hardware. It also indexes and keeps track of that data, enabling big-data processing and analytics far more effectively than was possible previously. Spark, on the other hand, is a data-processing tool that operates on those distributed data collections; it doesn't do distributed storage.

2. They do not depend upon each other
Hadoop includes not just a storage component, known as the Hadoop Distributed File System, but also a processing component called MapReduce, so you don't need Spark to get your processing done. Conversely, you can also use Spark without Hadoop. Spark does not come with its own file management system, though, so it needs to be integrated with one -- if not HDFS, then another cloud-based data platform. Spark was designed for Hadoop, however, so many agree they're better together.

3. When to use spark
MapReduce's processing style can be just fine if your data operations and reporting requirements are mostly static and

you can wait for batch-mode processing. But if you need to do analytics on streaming data, like from sensors on a factory floor, or have applications that require multiple operations, you probably want to go with Spark.

Conclusion

In order to abbreviate my findings I am compiling it in this matrix. This metric is only based on finite work with the different software packages and is not very precise. The divisions are:
GUI and graphics; ease of learning; Support

TABLE I.
MATRIX FORMAT FOR MULTIPLE TOOLS AVAILABLE ACCORDING TO MY FINDINGS.(OUT OF 5)

	Ease	Features	Design	Support
Weka	3	4	3	3
RapidMiner	2	3	3	3
Hadoop	3.7	3.8	4.3	3

TABLE II.
DIFFERENCE BETWEEN MULTIPLE BIG DATA MINING TOOLS.

	Hadoop	Apache Spark	RapidMiner
What it is	distributed data infrastructure	data-processing tool	a code-free environment for designing advanced analytic processes that push computations down to your Hadoop cluster.
Dependency	Default HDFS and MapReduce.	Can work with Hadoop/scala or any other equivalent algorithms.	Hive, MapReduce, Spark, Pig and Mahout, ensuring that each step in the predictive analytics process is correctly integrated and executed across core Big Data technologies.
Speed	Fast	Faster	Moderate
Processing	MapReduce operates in steps	operates on the whole data set in one fell swoop	Automatically creates and executes an optimal analytics plan for your Hadoop. It pushes analytic instructions into Hadoop and Spark where predictive analytics are executed across the entire cluster, taking advantage of the processing power of Hadoop
Ideal to use	Requirements are mostly static and you can wait for batch-mode processing. When data is really big.	analytics on streaming data, like from sensors on a factory floor, or have applications that require multiple operations	allows analysis upon the full breadth and variety your Big Data, as compared to solutions that can only analyze subsets and pieces of extracted Hadoop data.
Ease of coding	Writing hadoop mapreduce is complex and lengthy process.	Spark code is always compact than writing hadoop mapreduce.	No coding
	Disk based computing	RAM based Computing	Radoop Nests are used.

References

[1] http://www.ijarcsse.com/docs/papers/Volume_4/5_May2014/V4I5-0328.pdf
[2] http://www.itproportal.com/2013/12/20/big-data-5-major-advantages-of-hadoop/
[3] http://www.cs.waikato.ac.nz/ml/weka/bigdata.html
[4] http://thenewstack.io/six-of-the-best-open-source-data-mining-tools/
[5] https://en.wikipedia.org/wiki/RapidMiner
[6] http://thenewstack.io/six-of-the-best-open-source-data-mining-tools/
[7] http://dataconomy.com/hadoop-what-how-introduction/
[8] http://www.melekirmak.com/trepon/images/26022012152 1451.pdf
[9] http://shop.oreilly.com/product/0636920025122.do
[10] https://weka.wikispaces.com/
[11] https://sourceforge.net/
[12] https://rapidminer.com/products/radoop/
[13] https://www.dezyre.com/article/hadoop-mapreduce-vs-apache-spark-who-wins-the-battle/83
[14] http://research.ijcaonline.org/volume113/number1/pxc3900531.pdf