format, so that we can read it onto the Weka interface. After having done these courses, once has attained enough skills to start working and analyzing data sets using Weka GUI . Those who visited the MOOC link would have the seen the course 'More Data mining with Weka'.

Step 5: More Data Mining with Weka

Here, some more advanced features of using the software have been discussed. It builds up the experience from using the previous course, hence it's a prerequisite.

Besides this course you might want to have a look at this YouTube Lecture Series from Rushdi Shams.There are total of 38 lectures. You can skip a few of initial 2-3 lectures if you find the content to be similar to the courses above. This course has been built on various skills which are complementary to those provided by the above series.

There are some interesting discussions happening on Reddit about Weka. It is advisable to go through the mentioned link to gather news on Weka and how it is being used by others. This should give one enough perspective about next possible step after Weka.

Step 6: Weka Command Line

Next Step: As of now we have been relying on using Weka using Weka GUI. As of now both the courses rely on GUI for the purpose , those with experience in JAVA Programming can rely on calling Weka from within JAVA Code. This is useful because when trying or working out with large data sets scripting helps in automating your work. Also, since JAVA is used for Hadoop Framework, Weka can be used for BigData as well. You can read more using Weka in BigData from here.

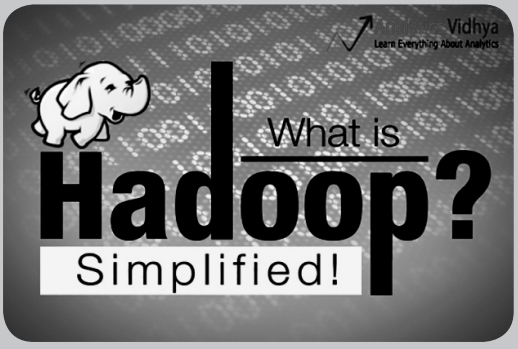
So, those interested in this aspect of Weka can try this lecture series by Dr Noureddin Sadawi. You may like to checkout this Weka API tutorial playlist also. The emphasis is on calling Weka API from within JAVA code, it repeats some of the above concepts but we use Weka using a command line interface.

Step 7: Word2Vec Challenge

Having gained significant insight, we will now have a look at sentiment analysis. There is a small data set with data sets size around 25 MB. So these can be processed using the Weka GUI. For data sets larger than 40 MB, we need to use the command line method. This discussion might be useful.

This path has been contributed by Abhinav Unnam, who interned with us last year. Abhinav is currently undergoing a dual degree course from IIT Roorkee, one of the India's finest Engineering Colleges. He started his machine learning journey through Weka and enjoys participating in several Kaggle competitions today using R and Kaggle.

What is Hadoop?



Scenario 1: Any global bank today has more than 100 Million customers doing billions of transactions every month

Scenario 2: Social network websites or e-Commerce websites track customer behavior on the website and then serve relevant information / product.

Traditional systems find it difficult to cope up with this scale at required pace in cost-efficient manner.

This is where Big data platforms come to help. In this article, we introduce you to the mesmerizing world of Hadoop. Hadoop comes handy when we deal with enormous data. It may not make the process faster, but gives us the capability to use parallel processing capability to handle big data. In short, Hadoop gives us capability to deal with the complexities of high volume, velocity and variety of data (popularly known as 3Vs).

Please note that apart from Hadoop, there are other big data platforms e.g. NoSQL (MongoDB being the most popular), we will take a look at them at a later point.

Introduction to Hadoop:

Hadoop is a complete eco-system of open source projects that provide us the framework to deal with big data. Let's start by brainstorming the possible challenges of dealing with big data (on traditional systems) and then look at the capability of Hadoop solution.

Following are the challenges I can think of in dealing with big data :

1. High capital investment in procuring a server with high processing capacity.

2. Enormous time taken

3. In case of long query, imagine an error happens on the last step. You will waste so much time making these iterations.

4. Difficulty in program query building

Here is how Hadoop solves all of these issues :

1. High capital investment in procuring a server with high processing capacity: Hadoop clusters work on normal commodity hardware and keep multiple copies to ensure reliability of data. A maximum of 4500 machines can be connected together using Hadoop.

2. Enormous time taken: The process is broken down into

pieces and executed in parallel, hence saving time. A maximum of 25 Petabyte (1 PB = 1000 TB) data can be processed using Hadoop.

3. In case of long query, imagine an error happens on the last step. You will waste so much time making these iterations : Hadoop builds back up data-sets at every level. It also executes query on duplicate datasets to avoid process loss in case of individual failure. These steps makes Hadoop processing more precise and accurate.

4. Difficulty in program query building: Queries in Hadoop are as simple as coding in any language. You just need to change the way of thinking around building a query to enable parallel processing.

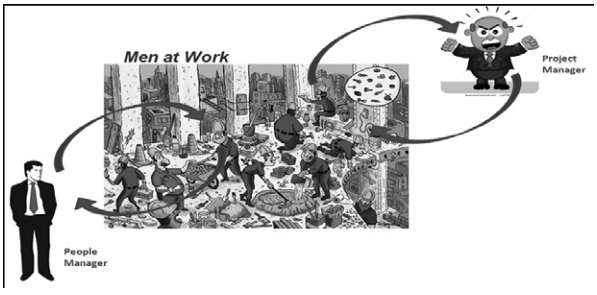
Background of Hadoop:

With an increase in the penetration of internet and the usage of the internet, the data captured by Google increased exponentially year on year. Just to give you an estimate of this number, in 2007 Google collected on an average 270 PB of data every month. The same number increased to 20000 PB everyday in 2009. Obviously, Google needed a better platform to process such an enormous data. Google implemented a programming model called MapReduce, which could process this 20000 PB per day. Google ran these MapReduce operations on a special file system called Google File System (GFS). Sadly, GFS is not an open source.

Doug cutting and Yahoo! reverse engineered the model GFS and built a parallel Hadoop Distributed File System (HDFS). The software or framework that supports HDFS and MapReduce is known as Hadoop. Hadoop is an open source and distributed by Apache.

Framework of Hadoop Processing:

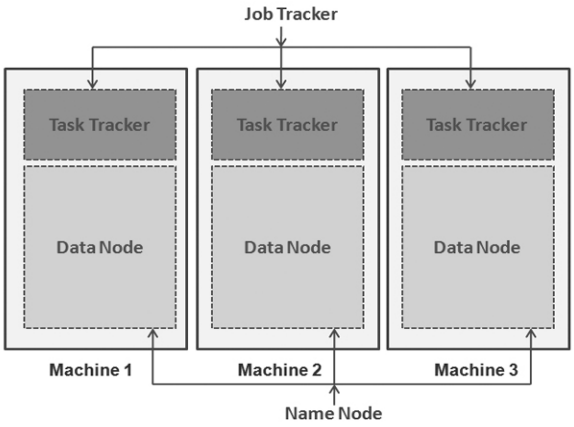
Let's draw an analogy from our daily life to understand the working of Hadoop. The bottom of the pyramid of any firm are the people who are individual contributors. They can be analyst, programmers, manual labors, chefs, etc. Managing their work is the project manager. The project manager is responsible for a successful completion of the task. He needs to distribute labor, smoothen the coordination among them etc. Also, most of these firms have a people manager, who is more concerned about retaining the head count.



Hadoop works in a similar format. On the bottom we have machines arranged in parallel. These machines are analogous to individual contributor in our analogy. Every machine has a data node and a task tracker. Data node is also known as HDFS (Hadoop Distributed File System) and Task tracker is also known as map-reducers.

Data node contains the entire set of data and Task tracker does all the operations. You can imagine task tracker as your arms and leg, which enables you to do a task and data node as your brain, which contains all the information which you want to process. These machines are working in silos and it is very essential to coordinate them. The Task trackers (Project manager in our analogy) in different machines are coordinated by a Job Tracker. Job Tracker makes sure that each operation is completed and if there is a process failure at any node, it needs to assign a duplicate task to some task tracker. Job tracker also distributes the entire task to all the machines.

A name node on the other hand coordinates all the data nodes. It governs the distribution of data going to each machine. It also checks for any kind of purging which have happened on any machine. If such purging happens, it finds the duplicate data which was sent to other data node and duplicates it again. You can think of this name node as the people manager in our analogy which is concerned more about the retention of the entire dataset.



When not to use Hadoop?

Till now, we have seen how Hadoop has made handling big data possible. But in some scenarios Hadoop implementation is not recommended. Following are some of those scenarios :

1. Low Latency data access : Quick access to small parts of data

2. Multiple data modification : Hadoop is a better fit only if we are primarily concerned about reading data and not writing data.

3. Lots of small files : Hadoop is a better fit in scenarios, where we have few but large files.

Summary:

This article gives you a view on how Hadoop comes to the rescue when we deal with enormous data. Understanding of the working of Hadoop is very essential before starting to code for the same. This is because you need to change the way of thinking of a code. Now you need to start thinking of enabling parallel processing. You can do many different types of processes on Hadoop, but you need to convert all these codes into a map-reduce function. In the next few articles we will explain how you can convert your simple logic to Hadoop based Map-Reduce logic. We will also take R-language specific case studies to build a solid understanding of the application of Hadoop.