As Per New Syllabus of VTU 2015 Scheme
Choice Based Credit System(CBCS)

ALL IN ONE SUNSTAR EXAM SCANNER B.E.

8TH Sem CSE/ISE

- *As Per VTU CBCS Pattern*
- *Three CBCS Model Question Papers (With Answers)*

- Internet of Things Technology
- Big Data Analytics
- Network Management
- System Modeling and Simulation

AUTHORED BY A TEAM OF EXPERTS

SUNSTAR PUBLISHER
#4/1, Kuppaswamy Building, 19th Cross,
Cubbonpet, Bangalore - 560002
Phone : 080 22224143
E-mail: sunstar884@gmail.com

CONTENTS

As Per New VTU Syllabus w.e.f 2015-16
Choice Based Credit System(CBCS)

SUNSTAR

SUNSTAR EXAM SCANNER

BIG DATA ANALYTICS

(VIII SEM. B.E. CSE / ISE)

SYLLABUS

BIG DATA ANALYTICS			
(AS PER CHOICE BASED CREDIT SYSTEM (CBCS) SCHEME) (EFFECTIVE FROM THE ACADEMIC YEAR 2016 - 2017)			
Subject Code	15CS82	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

MODULE 1

Hadoop Distributed File System Basics, Running Example Programs and Benchmarks, Hadoop MapReduce Framework, MapReduce Programming.

MODULE 2

Essential Hadoop Tools, Hadoop YARN Applications, Managing Hadoop with Apache Ambari, Basic Hadoop Administration Procedures

MODULE 3

Business Intelligence Concepts and Application, Data Warehousing, Data Mining, Data Visualization

MODULE 4

Decision Trees, Regression, Artificial Neural Networks, Cluster Analysis, Association Rule Minings.

MODULE 5

Text Mining, Naïve-Bayes Analysis, Support Vector Machines, Web Mining, Social Network Analysis.

Eighth Semester B.E. Degree Examination,
CBCS - Model Question Paper - 1
BIG DATA ANALYTICS

Time: 3 hrs. Max. Marks: 80

Note : *Answer any FIVE full questions, selecting ONE full question from each module.*

Module - 1

1. a. **Explain the Hadoop distribution file system design features with it's important aspects? (06 Marks)**

Ans. **Hadoop Distribution File System Design Features:** The Hadoop Distribution File System (HDFS) was designed for Big Data processing. The design assumes a large file write- once/ read many model that enables other optimizations and relaxes many of the concurrency and coherence overhead requirements of a true parallel file system. For instance, HDFS rigorously restricts data writing to one user at a time. All additional writers are "append-only", and there is no random writing to HDFS files. Bytes are always appended to the end of a stream, and byte streams are guaranteed to be stored in the order written.

The design of HDFS is based on the design of the Google Files System (GFS). A paper published by Google provides further background on GFS (http://research. google.com/ archive/gfs.html).

HDFS is designed for data streaming where large amounts of data are read from disk in bulk. The HDFS block size is typically 64MB or 128MB. Thus, this approach is entirely unsuitable for standard POSIX file system use. In addition, due to the sequential nature of the data, there is no local caching mechanism. The large block and file size make it more efficient to reread data from HDFS than to try cache the data.

The most interesting aspect of HDFS- and the one that separates it from other file system- is its data locality. A principal design aspect of Hadoop Map Reduce is the emphasis moving the computation to the data rather than moving the data to the computation. This distinction is reflected in how Hadoop will exist on hardware separate from the compute hardware. Data is then moved to and from the computer components via high-speed interfaces to the parallel file system array. HDFS, in contrast, is designed to work on the same hardware as the computer portion of the cluster. That is, a single server node in the cluster is often both a computation engine and a storage engine for the application.

Finally, Hadoop clusters assume node (and even rack) failure will occur at some point. To deal with this situation, HDFS has a redundant design that can tolerate system failure and still provide the data needed by the compute part of the program.

The following points summarize the important aspects of HDFS:

- The write-once/ read-many design is intended to facilitate streaming reads.
- Files may be appended, but random seeks are not permitted. There is no caching of data.
- Converged data storage and processing happen on the same server nodes.
- "Moving computation is cheaper than moving data".
- A reliable file system maintains multiple copies of data across the cluster. Consequently, failure of a single node (or even a rack in a large cluster) will not bring down the file system.
- A specialized file system is used, which is not designed for general use.

b. With a neat diagram explain various system roles in an HDFS deployment? (12 Marks)

Ans. HDFS Components

The design of HDFS is based on two types of nodes:

NameNode and multiple DataNodes.

In a basic design, a single NameNode manages all the metadata needed to store and retrieve the actual data from the DataNodes. No data is actually stored on the NameNode, however. For a minimal Hadoop installation, there needs to be a single NameNode daemon and single DataNode daemon running on at least one machine.

The design is a master/slave architecture in which the master (NameNode) manages the file system namespace and regulates access to files by clients. File system namespace operations such as opening, closing, and renaming files and directories are all managed by the NameNode. The NameNode also determines the mapping of blocks to DataNodes and handles DataNode failures.

The slaves (DataNodes) are responsible for serving read and write requests from the file system to the clients. The NameNode manages block creation, deletion, and replication.

An example of the client/NameNode/DataNode interaction is provide in Figure 1.1. When a client writes data, it first communicates with the NameNode and requests to crate a file. The NameNode determines how many blocks are needed and provides the client with the DataNodes that will store the data. As part of the storage process, the

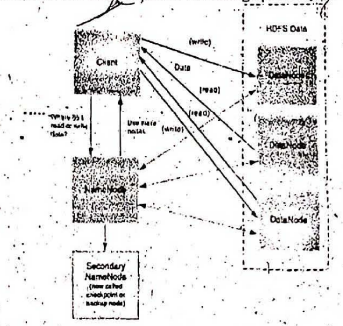


Figure 1.1 Various system roles in an HDFS deployment

data blocks are replicated after they are written to the assigned node. Depending on how many nodes are in the cluster, the NameNode will attempt to write replicas of the data blocks on nodes that are in other separate racks (if possible). If there is only one rack, then the replicated blocks are written to other servers in the same rack. After the DataNode acknowledges that the file block replication is complete, the client closes the file and informs the NameNode that the operation is complete. Note that the NameNode does not write any data directly to the DataNodes. It gives the client a limited amount of time to complete the operation. If it does not complete in the time period, the operation is canceled.

Reading data happens in a similar fashion. The client requests a file from the NameNode, which returns the nest DataNodes from which to read the data. The client then accesses the data directly from the DataNodes.

Thus, once the metadata has been delivered to the client, the NameNode steps back and

lets the conversation between the client and the DataNodes proceed. While data transfer is progressing, the NameNode also monitors the dataNodes by listening for heartbeats sent from DataNodes. The lack of a heartbeat signal indicates a potential node failure. In such a case, the NameNode will route around the failed DataNode and begin re-replicating the now-missing blocks. Because the file system is redundant, DataNodes can be taken offline (decommissioned) for maintenance by informing the NameNode of the DataNodes to exclude from the HDFS pool.

The mappings between data blocks and the physical DataNode are not kept in persistent storage on the NameNode.

For performance reasons, the NameNode stores all metadata in memory. Upon startup, each DataNode provides a block report (which it keeps in persistent storage) to the NameNode. The block reports are sent every 10 heartbeats. (The interval between reports is a configurable property.) The reports enable the NameNode to keep an up-to-data account of all data blocks in the cluster.

In almost all Hadoop deployments, there is a SecondaryNameNode. While not explicitly required by a NameNode, it is highly recommended. The term **"Secondary- NameNode" (now called CheckPointNode)** , it is not an active failover node and cannot replace the primary NameNode in case of its failure.

The purpose of the SecondaryNameNOde is to perform period checkpoints that evaluate the status of the NameNode. Recall that the NameNode keeps all system metadata memory for fast access. It also has two disk files that track changes to the metadata:

- An image of the file system state when the NameNode was started. This file begins with fsimage_* and is used only at startup by the NameNode.
- A series of modifications done to the file system after starting the NameNode. These file begin with edit_* and reflect the changes made after the fsimage_* file was read.

The location of these files is set by the dfs.namenode.name.dir property in the hdfs-site.xml file.

The SecondaryNameNode periodically downloads fsimage and edits files, joints them into a new fsimage, and uploads the new fsimage file to the NameNode. Thus, when the NameNode restarts, the fsimage file is reasonably up-to-data and requires only the edit logs to be applied since the last checkpoint. If the SecondaryNameNode were not running, a restart of the NameNode could take a prohibitively long time due to the number of changes to the file system.

OR

2. a. Explain the map reduce model with simple mapper script and simple reduce script. (08 Marks)

Ans. **The Map Reduce Model:** Apache Hadoop is often associated with Mapreduce computing. Prior to Hadoop version 2, this assumption was certainly true. Hadoop version 2 maintained the MapReduce capability and also made other processing models available to users. Virtually all the tools developed for Hadoop, such as Pig and Hive, will work seamlessly on top of the Hadoop version 2 MapReduce.

The MapReduce computation model provides a very powerful tool for many applications and is more common than most users realize. Its underlying idea is very simple.

There are two stages: a mapping stage and a reducing stage.

In the mapping stage, a mapping procedure is applied to input data. The map is usually some kind of filter or sorting process.

For instance, assume you need to count how many times the name "Kutuzov" appears in the novel War and Peace. One solution is to gather 20 friends and give them each a section of the book to search. This step is the map stage. The reduce phase happens when everyone is done counting and you sum the total as your friends tell you their counts.

Now consider how this same process could be accomplished using simple *nix command-line tools. The following grep command applied a specific map to a text file:

```
$ grep " Kutuzov " war-and-peace.txt
```

This command searches for the word Kutuzov (with leading and trailing space) in a text file called war-and-peace.txt. Each match is reported as a single line of text that contains the search term. The actual text file is a 3.2MB text dump of the novel War and Peace and is available from the book download page. The search term, Kutuzov, is a character in the book. If we ignore the grep count (-c) option for the moment, we can reduce the number of instances to a single number (257) by sending (piping) the results of grep into wc -l.

(wc -l or "word count" reports the number of lines it receives.)

```
$ grep "Kutuzov" war-and-peace.txt|wc -l
257
```

Though not strictly a MapReduce process, this idea is quite similar to and much faster than the manual process of counting the instances of Kutuzon in the printed book. The analogy can be taken a bit further by using the two simple (and naive) shell scripts shown in Listing 1.1 and Listing 1.2. The shell scripts are operation (much more slowly) and tokenize both the Kutuzov and Petersburg strings in the text:

```
$ cat war-and-peace.txt |./mapper.sh | ./reducer.sh
Kutuzov, 615
Petersburg, 128
```

Notice that more instance of Kutuzov have been found (the first grep command ignore instance like "Kutuzov." or "Kutuzov,"). The mapper inputs a text file and then outputs data in a (key, value) pair (token-name, count) format. Strictly speaking, the input to the script is the file and the keys are Kutuzov and Petersburg. The reducer script takes these key-value pairs and combines the similar token and counts the total number of instances. The result is a key-value pair (token-name, sum).

Listing 1.1 Simple Mapper Script'

```
#! /bin/bash
while read line ; do for token in $line; do
if ["$token"="Kutuzov"] ; then echo "Kutuzov,1"
elif ["$token" ="Petersburg"] ; then echo "Petersburg,1"
fi done done
```

Listing 1.2 Simple Reducer Script

```
#! /bin/bash kcount=0 pcount=0
while read line ; do
if ["$line"="Kutuzov,1"] ; then let kcount=kcount+1
elif ["$line" = "Petersburg,1"] ; then let pcount=pcount+1
done
echo "Kutuzov, $kcount" echo "Petersburg,$pcount"
```

Formally,the MapReduce process can be described as follows.

The mapper and reducer functions are both defined wrt data structured in (key,value) pairs.

The mapper takes one pair of the data with atype in one data domain,and returns a list of pairs in a different domain:

Map(key1,value1)-> list(key2,value2)

The reducer function is then applied to each key-value pair,which in turn produces a collection of values in the saim domain.

Reduce(key2 , list(value2))-> list(value3)

Each reducer call typically produces either one value (value3) or an empty response. Thus.the MapReduce framework transforms a list of (key,value) pairs into a list of values.

The MapReduce model is inspired by the map and reduce functions commonly used in many functional programming languages. The functional nature of MapReduce has some important properties:

i.Data flow is in one direction (map to reduce).It is possible to use output of a reduce step as the input to another MapReduce process.

ii. As with functional programming, the input data are not changed. By applying the mapping and reduction functions to the input data, new data are produced. In effect,the original state of Hadoop data lake is always preserved.

iii. Because there is no dependency on how the mapping and reducing functions are applied to the data,the mapper and reducer data flow can be implemented in any number of ways to provide better performance.

Distributed(parallel) implementations of MapReduce enable large amounts of data to be analyzed quickly. In general, the mapper process is fully scalable and be applied to any subset of the input data. Results from multiple parallel mapping functions are then combined in the reducer phase.

b. Explain compiling and running process of the Hadoop word count example with program. (08 Marks)

Ans. WordCount is a simple application that counts the number of occurrences of each word in a given input set. The MapReduce framework operates exclusively on key- value pairs; that is, the framework views the input to the job as a set of key-value pairs and produces a set of key-value pairs of different types. The MapReduce job proceeds as follows:

(input) <k1, v1> -> map -> <k2, v2> -> Combine -> <k2, v2> reduce -> <k3, v3> (output)

The mapper implementation,.via the map method, processes one line at a time as provided by the specified TextInputFormat class. It then splits the line into tokens separated by whitespaces using the StringTokenizer and emits a key-value pair of<word, 1>. The relevant code section is as follows:

```
public void map(object kay, Text value, Context context
) throws IOException, InterruptedException { StringTokenizer itr=new StringTokenizer
(value.tostring ( ) ) ; while (itr.hasMoreTokens ( ) ) ;
word.set (itr.nexToken ( ) ) ; context.write(word, one) ;
}
}
```

Given two input files with contents Hello World Bye World and Hello Hadoop Goodbye Hadoop, the WordCount mapper will produce two maps:

```
<Hello, 1>
<World, 1>
<Bye, 1>
<World, 1>
<Hello, 1>
<Hadoop, 1>
```

<Goodbye, 1>
<Hadoop, 1>
WordCount sets a mapper
job.setMapperClass (TokenizerMapper.class) ;
a combiner job.setCombinerClass(IntSumReducer.class) ;
and a reducer job.setCombinerClass(IntSumReducer class) ;
Hence, the output of each map is passed through the local combiner (which sums the values in the same way as the reducer) for local aggregation and then sends the data on to the final reducer. Thus, each map above the combiner performs the following pre-reductions:
<Bye, 1>
<Hello, 1>
<World, 2>
<Goodbye, 1>
<Hadoop, 2>
<Hello, 1>
The reducer implementation, via the reduce method, simply sums the values, which are the occurrence counts for each key. The relevant code section is as follows: public void reduce
(Text key, Iterable<IntWritable> values,
Context context
) throws IOException, InterruteException { int sum=0;
for (IntWritable val : values) { sum+= val.get ( ) ;
}
result.set (sum); context.write (key, result) ;
}
The final output of the reducer is the following:
<Bye, 1>
<Goodbye, 1>
<Hadoop, 2>
<Hello, 2>
<World, 2>
To compile and run the program from the command line, perform the following steps:
1. Make a local wordcount_Classes directory.
$ makdir wordcount_classes
2. Compile the WordCount.java program using the 'hadoop classpath' command to include all the available Hadoop class paths.
$ javac -cp 'hadoop classpath' -d wordcount_classes WordCount.java
3. The jar file can be created using the following command:
$ jar -cvt wordcount.jar -C wordcount_classes/
4. To run the example, create an input directory in HDFS and place a text file in the new directory. For this example, we will use the war-and-peace.txt:
$ hdfs dfs -mkdir war-and-peace-input
$ hdfs dfs -put war-and-peace.txt war-and-peace-input
5. Run the WordCount application using the following command:
$ hadoop jar wordcount.jar WordCount war-and-peace-input
→ war-and-peace-output
If everything is working correctly, Hadoop messages for the job should look like the following (abbreviated version):

15/05/24 18:13:26 INFO impl.TimelineclientImp l: Timeline service address: http:// limulus:8188/ws/v1/timeline/
15/05/24 18.13:26 INFO client .RMProxy: Connecting to ResourceManager at limulus/10 0.0.1:8050
15/05/24 18:13:26 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
15/05/24 18:13:26 INFO input.FileInputFormat : Total input paths to process ; 1 15/05/24 18:13:27 INFO mapreduce.JobSubmitter: number of splits:1
[...]
File Input Format Counters
Bytes Read=3288746
File Output Format Counters.
Bytes Written=467839
In addition, the following files should be in the war-and-peace-output directory. The actual file name may be slightly different depending on your Hadoop version.
$ hdfs dfs -ls war-and-peace-output
Found 2 times
-rw-r--r-- 2 hdfs hdfs 0 2015-05-24 11:14 war-and-peace-output / _SUCCESS
-rw-r--r-- 2 hdfs hdfs 4678639 2015-05-24 11:14 war-and-peace-output/ part -r-00000
The complete list of word counts can be copied from HDFS to the working directory with the following command:
$ hdfs dfs -get war-and-peace-output/part -r-00000
If the WordCount program is run again using the same outputs, it will fail when it tries overwrite the war-and-peace-output directory. The output directory and all contents can be removed with the following command:
$ hdfs dfs -rm -r -skipTrash war-and-peace-output

Module -2

3. a. Explain with example Apache pig and Apache Hive? (08 Marks)

Ans. Apache Pig is a high-level language that enables programmers to write complex MapReduce transformations using a simple scripting language. Pig Latin (the actual language) defines a set of transformations on a data set such as aggregate, join, and sort. Pig is often used to extract, transform, and load (ETL) data pipelines, quick research on raw data, and iterative data processing.

Apache Pig has several usage modes. The first is a local mode in which all processing is done on the local machine. The non-local (cluster) modes are MapReduce and Tez. These modes execute the job on the cluster using either the MapReduce engine or the optimized Tez engine.

Table 2.1 Apache Pig Usage Modes

	Local Mode	Tez Local Mode	Map Reduce Mode	Tez Mode
Interactive Mode	Yes	Experimental	Yes	Yes
Batch Mode	Yes	Experimental	Yes	Yes

There are also interactive modes, using small amounts of data, and then run at developed locally in interactive modes, using small amounts of data, and then run at scale on the cluster in a production mode. The modes are summarized in Table 2.1.

Pig Example Walk-Through

For this example, the following software environment is assumed. Other environments should work in a similar fashion.

- OS: Linux
- Platform: RHEL 6 6
- Hortonworks HDP 2.2 which Hadoop version: 2.6
- Pig version: 0.14.0

If pseudo-distributed installation is used, "Installation Recipes," instructions for installing Pig are .

In this simple example, Pig is used to extract user names fromt he /etc /passwd file. A full description of the Pig Latin language is beyond the scope of this introduction, but more information about Pig can be found at http://pig.apache,org/docs/r0.14.0/ start.html. The following example assumes the user is hdfs,but any valid user with access to HDFS can run the example.

To begin the example, copy the passwd file to a working directory for local Pig operation:

```
$ cp /etc/passwd
```

Next, copy the data file into HDFS for Hadoop MapReduce operation:

```
$ hdfs dfs -put passwd passwd
```

You can confirm the file is in HDFS by entering the following command:

```
hadfs dfs -ls passwd
-rw-r--r-- 2 hdfs hdfs 2526 2015-03-19 11:08 passwd
```

In the following example of local Pig operation, all processing is done on the local machine (Hadoop is not used). First, the interactive command line is started:

```
$ pig -x local
```

If Pig starts correctly, you will see a grunt> prompt. You may also see a bunch of INFO messages, which you can ignore. Next, enter the following commands to load the passwd file and then grab the user name and dump it to the terminal. Note that Pig commands must end with a semicolon (;).

```
grunt> A = load 'passwd' using PigStorage (':');
 grunt> B = foreach A generate $0 as id;
grunt> dump B;
```

The processing will start and a list of user names will be printed to the screen. To exit the interactive session, enter the command quit.

```
$ pig -x mapreduce
```

The same sequence of commands can be entered at the grunt> prompt. You may wish to change the $0 argument to pull out other items in the passwd file. In the case of this simple script,you will notice that the MapReduce version takes much longer. Also, becuase we are running this application under Hadoop, make sure the file is placed in HDFS.

If you are using the Hortonworks HDP distribution with tez installed, the tez engine can be used as follows:

```
$ pig -x tez
```

Pig can also be from a script. An example script (id.pig) is available from the example code download (see Appendix A, "Book Webpage and Code Download"). This script, which is repeated here,is designed to the same things as the interactive version:

```
/* id.pig */
A = load 'passwd' using PigStorage (':'); -- load the passwd file
B = foreach A generate $0 as id; -- extract the user IDs
dump B;
store B into 'id.out'; --write the results to a directory name id.out
```


Comments are delineated by /*/* and -- at the end of a line. The script will create a directory called id.out for the results. First, ensure that the id out directory is not in your local directory, and then start Pig with the script on the command line:

```
$ /bin/rm -r id.out/
$ pig -x local id.pig
```

If the script worked correctly, you should see at least one data file with the results and a zero-length file with the name_SUCCESS. To run the MapReduce version, the same procedure; the only difference is that now all reading and writing taken place in HDFS.

```
$ hdfs dfs -rm -r id.out
$ pig id:pig
```

If Apache tez is installed, you can run the example script using the -x tez option. You can learn more about writing Pig script at http://pig.apache.org/docs/r0.14.0/start. html.

Using Apache Hive

Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, ad hoc queries,and the analysis of large data sets using a SQL- like language called HiveQL. Hive is considered the de facto standard for interactive SQL queries over petabytes of data using Hadoop and offers the following features:

- Tools to enable easy data extraction, transformation, and loading (ETL)
- A mechanism to impose structure on a variety of data formats
- Access to files stored either directly in HDFS or in other data storage systems such as HBase
- Query execution via MapReduce and Tez (optimized MapReduce)

Hive provides users who are already familiar with SQL the capability to query the data on Hadoop clusters. At the same time, Hive makes it possible for programmers who are familiar with the MapReduce framework to add their custom mappers and reducers to Hive queries. Hive queries can also be dramaticallly accelerated using the Apache Tez framework usnder YARN in Hadoop version 2.

Hive Example Walk-Through

For this example, the following software environment is assumed. Other environments should work in a similar fashion.

- OS: Linux
- Platform: RHEL 6.6
- Hortonworks HDP 2.2 with Hadoop version: 2.6
- Hive version: 0.14.0

Although the following example assumes the user is hdfs, any valid user with access to HDFS can run the example.

To star Hive, simply enter the hive command. If Hive starts correctly, you should get a hive> prompt.

```
$ hive
(some message may show up here)
hive>
```

As a simple test, create and drop a table. Note that Hive commands must end with a semicolon (;).

```
hive> CREATE TABLE pokes (foo INT, bar STRING)
OK
Time taken: 1.705 seconds
Hive>.SHOW TABLES;
```

```
OK
pokes
Time taken: 0.174 seconds, Fetched: 1 row (s)
hive> DROP TABLE pokes
OK
Time taken: 4 038 seconds
```

A more detailed example can be developed using a web server log file to summarize message types. First, create a table using the following command:

```
hive> CREATE TABLE logs(t1 string, t2 string, t3 string, t4 string, '
→ t5 string, t6 string, t7 string) ROW FORMAT DELIMITED FIELDS
→ TERMINATED BY ' ';
OK
Time taken: 0.129 seconds
```

Next, load the data-in this case,from the sample.log file. This file is available from the example code download. Note that file is found in the local directory and not in HDFS.

```
hive> LOAD DATA LOCAL INPATH 'sample.log' OVERWRITE INTO TABLE logs;
Loading data to table default .logs
Table default.logs stats: (numFiles=I, numRow=0, totalSize=99271, rawDataSize=0)
OK
Time taken: 0.953 seconds
```

Finally, apply the select step to the file. Note that this invokes a Hadoop MapReduce operation. The results appear at the end of the output (e.g., totals for the message types DEBUG, ERROR, and so on).

```
hive> SELECT t4 AS sev, COUNT(*) AS cnt FROM logs WHERE t4 LIKE '(%' GROUP
BY t4;
Query ID = hdfs_20150327130000_d1e1a265-a5d7-4ed8-b785-2c6569791368 Total jobs = I
Launching Job I out of I
Number of reduce tasks not specified. Estimated from input data size: I
In order to change the average load for a reducer (in bytes) :
set hive .exec.reduce.bytes.per.reducer=<number>
 In order to limit the maximum number of reducers:
 set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
set mapreduce.job.reduces=<number>
Starting Job = job_1427397392757_0001, Tkeing URL = http://norbert:8088/proxy/
application_142739739257_0001/
Kill Command =/opt/hadoop-2.6.0/bin/hadoop job -kill job_1427397392757_0001 Hadoop
job information for Stage-I: number of mappers: I; number of reducers: I 2015-03-27
13:00:17,399 Stage-I map= 0%, reduce = 0%
2015-03-27 13:00:26.100 Stage-I map= 100%, reduce = 0%, Cumulative CPU 2.14 sec
2015-03-27 13:00:34,979 Stage-I map= 100%, reduce = 100%, Cumulative CPU 4.07 sec
MapReduce Total cumulative CPU time: 4 seconds 70 msec
Ended Job = job_1427397392757_0001
MapReduce Jobs Launched:
Stage-Stage-I: Map: I Reduce: I Cumulative CPU: 4.07 sec HDFS Read: 106384
HDFS Write; 63 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 70 msec
OK
```



```
[DEBUG] 434
[ERROR] 3
[FATAL] I
[INFO] 96
[TRACE] 816
[WARN] 4
Time taken: 32.624 seconds, Fetched: 6 row(s)
To exit Hive, simply type exit; :
hive> exit;
```

b. **Explain with the following commands in the H base data model.** **(04 Marks)**

1) Create the database
2) Inspect the database
3) Create row
4) Delete a row
5) Remove a table
6) Adding data in Bulk.

Ans. **1) Create the Database**

The next step is to create the database in HBase using the following command:

```
hbase (main) :006:0> create 'apple', 'price', 'volme'
0 row(s) in 0.8150 seconds
```

In this case, the table name is apple, and two columns are defined. The data will be used as the row key. The price column is a family of four values (open, close, low, high). The put command is used to add to the database from within the shell. For instance, the preceding data can be entered by using the following commands:

```
put 'apple', '6-May-15', 'price:open', '126.56'
put 'apple', '6-May-15', 'price:high', '126.75'
put 'apple', '6-May-15', 'price:low', '125.36'
put 'apple', '6-May-15', 'price:close', '125.01'
put 'apple', '6-May-15', 'volume', '71820387'.
```

Note that these commands can be copied and pasted into HBase shell and are available from the book download files . The shell also keeps a history for the section, and previous commands can be retrieved and edited for resubmission.

2) Inspect the Database

The entire database can be listed using the scan command. Be careful when using this command with large database. This example is for one row.

```
scan 'apple'
hbse (main) :006:0> scan 'apple'
Row        COLUMN+CELL
6-May-15 column=price:close, timestamp=1430955128359, value=125.01
6-May-15 column=price:high, timestamp=1430955126024, value=126.75
6-May-15 column=price:low, timestamp=1430955126053, value=123.36 6-May-15
column=price:open, timestamp=1430955125977, value=126.56
6-May-15 column=volume:, timestamp=1430955141440, value=71820387
```

3) Get a Row

You can use the row key to access an individual row. In the stock price database, the data is the row key.

```
hbase (main) :008:0> get 'apple', '6-May-15'
  COLUMN          CELL
price:closetimestamp=1430955128359, value=125.01
price:high timestamp=1430955126024, value=126.75
price:low  timestamp=1430955126053, value=123.36
price:open timestamp=1430955125977, value=126.56
volume:    timestamp=1430955141440, value=71820387
row(s) in 0.130 seconds
```

4)Delete a Row

You can delete an entire row by giving the deleteall command as follows:

```
hbase(main) :009:0> deleteall 'apple', '6-May-15'
```

5) Remove a Table

To remove (drop) a table, you must first disable it. The following two commands remove the apple table from Hbase: hbase(main) :009:0> disable 'apple' hbase(main) :010:0> drop 'apple'

6)Adding data in Bulk

There are several ways to efficiently load bulk data into HBase. Covering all of these methods is beyond the scope of this chapter. Instead, we will focus on the ImportTsv utility, which loads data in tab-separated values (tsv) format into HBase. It has two distinct usage modes:

- Loading data from a tsv-format file is HDFS into HBase via the put command
- Preparing StoreFiles to be loaded via the completebulkload utility

The following example shows how to use ImporTsv for the first option, loading the tsv-format file using the put command. The second option works in a two-step fashion and can be explored by consulting http://hbase.apache.org/book.html#importtsv.

The first step is convert the Apple-stock.csv file to tsv format. The following script, which is included in the book software, will remove the first line and do the conversion. In doing so, it creates a file named Apple-stock.tsv.

```
$ convert-to-tsv.sh Apple-stock.tsv /tmp
```

Finally, ImportTsv is run using the following command line. Note the column designation in the -Dimporttsv.columns option. In the example, the HBASE_ROW_ KEY is set as the first column-that is, the data for the data.

```
$ hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.columns=
→ HBASE_ROW_KEY, price:open,price:high,price:low,price:close,volume
→ apple /tmp/Apple-stock.tvs
```

The ImportTvs command works will use MapReduce to load the data into HBase. To verify that the command works, drop and re-create the apple database, as described previously, before running the import command.

c. **What is YARN? Explain any five commands?** **(04 Marks)**

Ans. The Hadoop YARN project includes the Distributed-Shell application, which is an example of a Hadoop non-MapReduce application built on top of YARN. Distributed- Shell is a simple mechanism for running shell commands and scripts in containers on multiple nodes in a Hadoop cluster. This application is not meant to be a production administration tool, but rather a demonstration of the non-MapReduce capability that can be implemented on top of YARN. There are multiple mature implementations of a distributed shell that administrators typically use to manage a cluster of machines. In addition, Distributed-Shell can be used as a starting point for exploring and building Hadoop YARN applications. This chapter offers guidance on how the Distributed-Shell can be used to understand the operation of YARN applications.

Using the YARN Distributed-Shell

For the purpose of the examples presented in the remainder of this chapter. we assume and assign the following installation path, based on Hortonworks HDP 2.2, the Distributed-Shell application:

```
$ export YARN_DS=/usr/hdp/current/hadoop-yarn-client/hadoop-yarn-applications-distributedshell.jar
```

For the pseudo-distributed install using Apache Hadoop version 2.6.0, the following path will run the Distributed-Shell application (assuming $HADOOP_HOME is defined to reflect the location Hadoop):

```
$ export YARN_DS=$HADOOP_HOME/share/hadoop/yarn/hadoop-yarn-applications-distributedshell-2.6.0.jar
```

If another distribution is used, search for the file hadoop-yarn-applications- distributedshell*. jar and set $YARN_DS based on its location. Distributed-Shell exposes verious options that can be found by running the following command:

```
$ yarn org.apache.hadoop.yarn.applications.distributedshell.Client -jar $YARN_DS
→ -help
```

The output of this command follows:

```
usage: client
```

-appname <arg>	Application Name. Default value -distributedShell
-attempt_failures_validity_ interval <arg>	when attempt_failure_validity_ interval in milliseconds is set to >0, the failures number will not take failure which happen out of the validityInterval into failure count. If failure count reaches to maXAppAttempts, the application will be failed.
-container_memory <arg>	Amount of memory in MB to be requested to run the shell command
-container_vcores <arg>	Amount of virtual cores to be requested to run the shell command
-create	Flag to indicate whether to create the domain specified with -domain.
-debug	Dump out information
-domain <arg>	ID of the timeline domain where the timeline entities will be put
-help	Print usage
-jar <arg>	Jar file containing the application master
-keep_containers_across_ application_attempts	Flag to indicate whether to keep containers across application attempts. If the flag is true. running containers will not be retrieved by the new application attempt.
-log_properties <arg>	log4j.properties file
-master_memeory <arg>	Amount of memory in MB to be requested to run the. application master
-master_vcores <arg>	Amount of virtual cores to be requested to run the application master

-modify_acls <arg>	Users and groups that allowed to modify the timline entities the timeline entities in the given domain
-node_label_expression <arg>	Node label expression to determine the nodes where all the containers of this application will be allocated, " " means containers can be allocated anywhere,if you don't specify the option, default node_label_expression of queue will be used.
-num_containers <arg>	No. of containers on which the shell command needs to be executed
-priority <arg>	Application Priority. Default 0
-queue <arg>	RM Queue in which this application is to be submitted
-shell_args <arg>	Command line arge for the shell script. Multiple args can be separated by empty space.
-shell_cmd_priority <arg>	Priority for the shell command containers
-shell_command <arg>	Shell command to be executed by the Application Master. Can only specify either - -shell_command or - -shell_script
-shell_env <agr>	Environment for shell script. Specified asenv_ key=env_ val pairs
-shell_script <arg>	Location of the shell script to be executed. Can only specify either - -shell_command or - -shell_script
-timeout <arg>	Application timeout in milliseconds
-view_acls <arg>	Users and group that allowed to view the timeline entities in the given domain

OR

4. n. **Explain virus of Apache Ambari?** **(12 Marks)**

Ans. After completing the initial installation and logging into Ambari) a dashboard similar to that shown in Figure 4.1 is presented. The same four-node cluster as created that will be used to explore Ambari . It you need to reopen the Ambari dashboard interface, simply enter the following command (which assumes you are using the Firefox browser, although other browsers may also be used):

$ firefox localhost : 8080

The default login and password are admin and admin, respectively. Before continuing any further, you should change the default password. To change the password, select Manage Ambari from the Admin pull-down menu in the upper-right corner. In the management window, click Users under User + Group Management, and then click the admin user name. Select Change Password and enter a new password. When you are finished, click the Go To Dashboard link on the left side of the window to return to the dashboard view.

To leave the Ambari interface, select the Admin pull-down menu of the installed services. A glance at the dashboard should allow you to get a sense of how the cluster is performing. The top navigation menu bar, shown in figure 4.1, provides access to the Dashboard, Services, Hosts, Admin and Views features (the 3 × 3 cube is the Views menu). The status (up/down)

of various Hadoop service is displayed on the left using green/orange dots. Note that two of the service managed by Ambari are Nagios and Ganglia; the standard cluster management services managed by Ambari, they are sued to provide cluster monitoring (Nagios) and metrics (Ganglia).

Dashboard View

The Dashboard view provides small status widgets for many of the service running on the cluster. The actual services are listed on the left-side vertical menu. You can move, edit, remove, or add these widgets as follows:

- Moving: Click and hold a widget while it is moved about the grid.
- Edit: Place the mouse on the widget and click the gray edit symbol in the upper- right corner of the widget. You can change several different aspects (including thresholds) of the widget.
- Remove: Place the mouse on the widget and click the X in the upper-left corner.
- Add: Click the small triangle next to the Mterics tab and select Add. The available widgets will be displayed. Select the widgets you want to add and click Apply.

Some widgets provide additional information when you move the mouse over them. For instance, the DataNodes widget displays the number of live, dead, and view. For instance, Figure 4.2 provides a detailed view of the CPU Usage widget from Figure 4.1.

The Dashboard view also includes a heatmap view of the cluster. Cluster heatmaps physically map selected metrics across the cluster. When you click the Heatmaps tab, a heatmap for the cluster will be displayed. To select the metric used for the heatmap, choose the desired option from the Select Metric pull-down menu. Note that the scale ory used is displayed in Figure 4.3

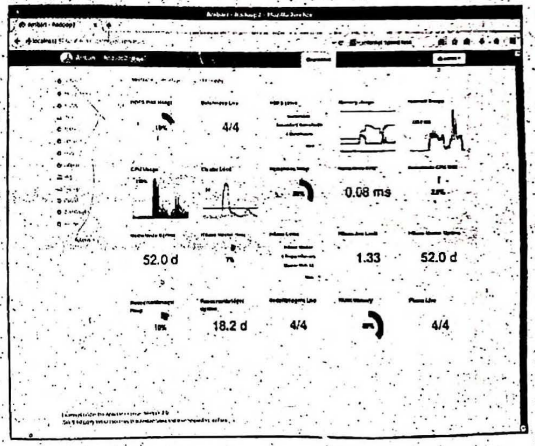


Figure 4.1 Apache Ambari dashboard view of a Hadoop Cluster

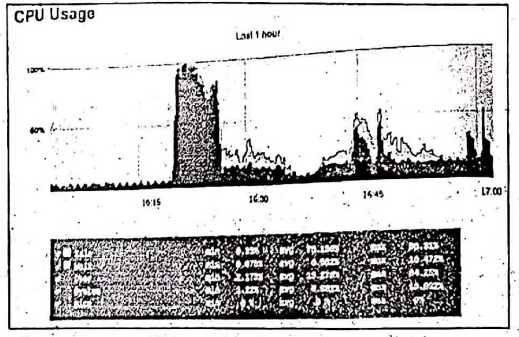


Figure 4.2 Enlarged view of Ambari CPU Usage widget

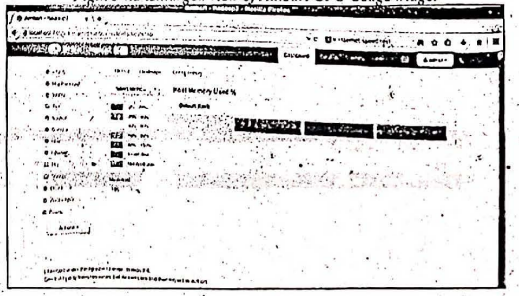


Figure 4.3 Ambari heatmap for Host memory usage

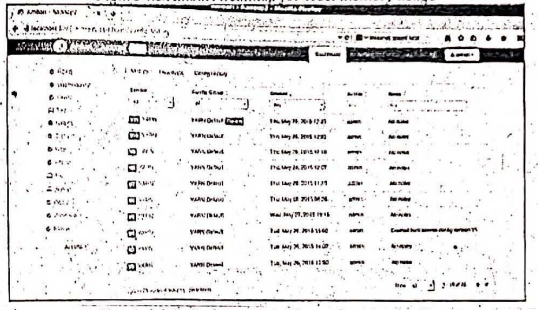


Figure 4.4 Ambari master configuration changes list

Configuration history is the final tab in the dashboard window. This view provides a list of configuration changes made to the cluster. As shown in Figure 4.4, Ambari enable configurations to be sorted by service, configuration, group, data, and author. To find the specific configuration settings, click the service name. More information on configuration setting is provided later in the chapter.

Service View

The service menu provides a detailed look at each service running on the cluster. It also provides a graphical method for configuring each service (i.e., instead of hand-editing the / etc/hadoop/conf XML files). The summary tab provides a current Summary view of important service metrics and an Alters and Health Checks sub- window.

Similar to the Dashboard view, the currently installed services are listed on the left- side menu. To select a service, click the service name in the menu. When applicable, each service will have is own summary, Alters and Health Monitoring and Service Metrics windows. For example, Figure 4.5 shows the service view for HDFS. Important information such as the status of NameNode, SecondaryNameNode, DataNodes, uptime, and available disk space is displayed in the Summary window. The Alters and Health Checks widow provides the latest status of the service and its component systems. Finally, several important real-time service metrics are displayed as widgets at the bottom of the screen.

As on the dashboard, these widgets can be expanded to display a more detailed view. Clicking the Configs tab will open an options from, shown in Figure 4.6, for the service. The options (properties) are the same ones that are set in the Hadoop XML should manage them only through the Ambari interface-that is, the user should not edit the files by hand.

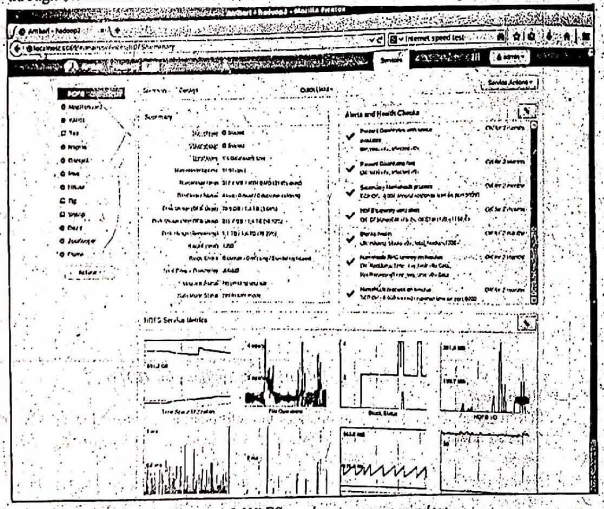


Figure 4.5 HDFS service summary window

The current settings available for each service are shown in the form. The administrator can set each of these properties by changing the values in the form. Placing the mouse in the input box of the property displays a short description of each property. Where possible, property are grouped by functionality. The form also has provisions for adding properties that are not listed. An example of changing service properties and restarting the service components is provided in the "Managing Hadoop Service" section.

If a service provides its own graphical interface (e.g., HDFS, YARN, Oozie), then that interface can be opened in a separate browser tab by using the Quick Links pull- down menu located in top middle of the window.

Finally, the Service Action pull-down menu in the upper-left corner provides a method for starting and stopping each service and/or its component daemons across the cluster. Some service may have a set of unique actions (such as rebalancing HDFS) that apply to only certain situations. Finally, every service has a Service Check option to make sure the service is working property. The service check is initially run as part of the installation process and can be valuable when diagnosing problems.

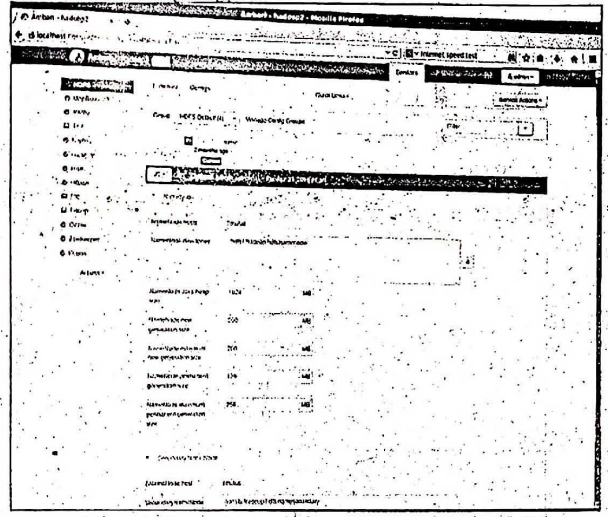


Figure 4.6 Ambari service options for HDFS

Hosts View

Selecting the Hosts menu item provides the information shown in Figure 4.7. The host name, IP address, number of cores, memory, disk usage, current load average, and Hadoop components are listed in this window in tabular form.

To display the Hadoop components installed on each host, click the links in the rightmost columns. You can also add new hosts by using the Actions pull-down menu. The new host must be running the Ambari agent (or the root SSH key must be entered) and have the base

software installed. The remaining options in the Actions pull-down menu provide control over the various service components running on the hosts.

Further details for a particular host can be found by clicking host name in the left column. As shown in Figure 4.8, the individual host view provides three sub- windows: Components, Host Metrics, and Summary information. The Components window lists the services that are currently running on the host. Each service can be stopped, restarted, decommissioned, or placed in maintenance mode. The Metrics window displays widgets that provide important metrics (e.g., CPU, memory, disk, and network usage). Clicking the widget displays a larger version of the graphic. The Summary window provides basic information about the host, including the last time a heartbeat was received.

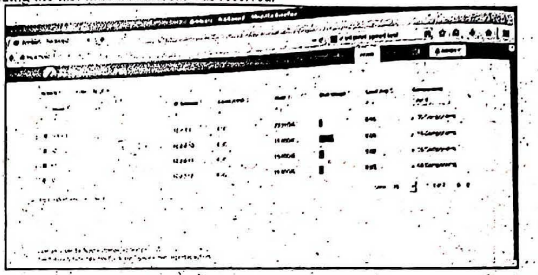


Figure 4.7 Ambari main Hosts screen

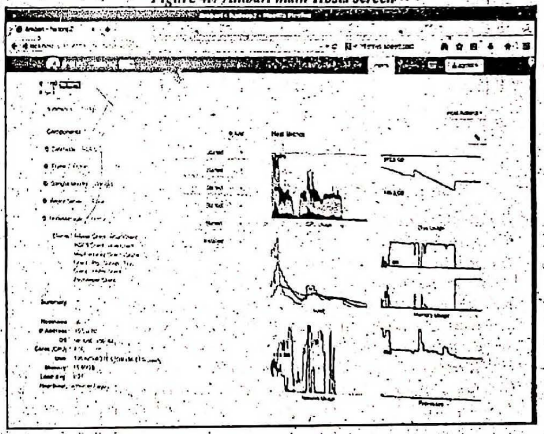


Figure 4.8 Ambari cluster host detail view

Admin View

The Administration (Admin) view provides three options. The first, as shown in Figure 4.9, displays a list of installed software. This repositories listing generally reflects the version of

Hortonworks Data Platform (HDP) used during the installation process. The Service Accounts option lists the service accounts added when the system was installed. These accounts are used to run various service and tests for Ambari. The third option, Security, sets the security on the cluster. A fully secured Hadoop cluster is important in many instances and should be explored if a secure environment is needed.

Views view

Ambari Views is a framework offering a systematic way to plug in user interface capabilities that provide for custom visualization, management, and monitoring features in Ambari. Views allows you to extend and customize Ambari to meet your specific needs.

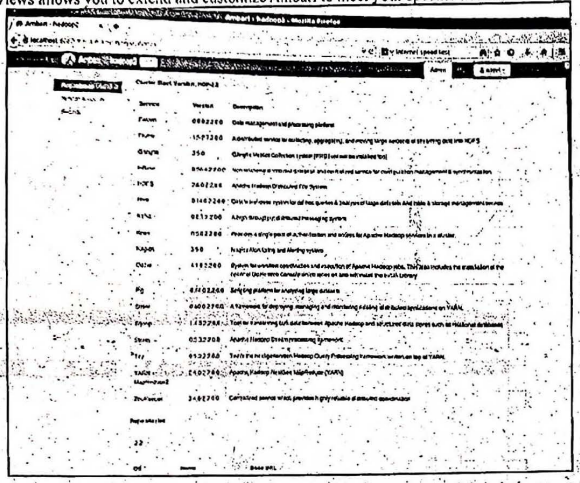


Figure 4.9 Ambari installed package with versions,numbers and descriptions

b. **Explain the Basic Hadoop YARN administration?** **(04 Marks)**

Ans. YARN has several administrative features and commands. To find out more about them, examine the YARN commands documentations at https://hadoop.apache.org/ docs/current/ hadoop-yarn/hadoop-yarn-site/YarnCommands.html#Administration_ Commands. The main administration commands is yarn rmadmin (resource manager administration). Enter yarn rmadmin -help to learn more about the various options.

Decommissioning YARN Nodes

If a NodeManager host/nodes to be removed from the cluster, it should be decommissioned first. Assuming the node is responding, you can easily decommission it from the Ambari web UI. Simply go to the Hosts view, click on the host, and select Decommission from the pull-down menu next to the NodeManager component. Node that the host may also be acting as a HDFS DataNode. Use the Ambari Hosts view to decommission the HDFS host in a similar fashion.

YARN WebProxy

The Web Application Proxy is a separate proxy server in YARN that addresses security issues with the cluster web interface on ApplicationMasters. By default, the proxy runs as part of the Resource Manager itself, but it can be configured to run in a stand-alone mode by adding the configuration property yarn.web-proxy.address to yarn-site.xml. (Using Ambari, go to the YARN Configs view, scroll to the bottom, and select Custom yarn-site.xml/Add property.) In stand-alone mode, yarn.web- proxy.principle and yarn.web-proxy.keytab control the Kerberos principal name and the corresponding keytab, respectively, for use in secure mode. These elements canbe added to the yarn-site.xml if required.

Using the JobHistoryServer

The removal of the JobTracker and migration of MapReduce from a system to an application-level framework necessitated creation of a place to store MapReduce job history. The JobHistoryServer provides all YARN MapReduce applications with a central location in which to aggregate completed jobs for historical reference and debugging. The settings for the JobHistoryServer can be found in the mapred-site. xml file.

Managing YARN Jobs

YARN jobs can be managed using the yarn application command. The following options, including -kill, -list, and -status, are available to the administrator with this command. MapReduce jobs can also be controlled with the mapred job command. usage: application

-appTypes <comma-separated list of application types> Works with ----list to filter applications based on their type.

-help Displays help for all commands.

-kill <Application ID> Kills the application.

-list Lists applications from the RM. Supports optional use of appTypes to filter applications based on application type.

-status <Application ID> Prints the status of the application.

Neither the YARN ResourceManager UI nor the Ambari UI can be used to kill YARN applications. If a job needs to be killed, give the yarn application command to find the Application ID and then use the -kill argument.

Setting Container Memory

YARN manages application resource containers over the entire cluster. Controlling the amount of container memory takes place through three important values in the yarn-site.xml file:

- yarn.nodemanager.resource.memory-mb is the amount of memory the NodeManager can use for containers.
- scheduler.minimum-allocation-mb is the smallest container allowed by the ResourceManager. A requested container smaller than this value will result in an allocated container of this size (default 1024MB).
- yarn.scheduler.maximum-allocation-mb is the largest container allowed by the ResourceManager (default 8192MB).

Setting Container Cores

You can set the number of cores for containers using the following properties in the yarn-site. xml:

- yarn.scheduler.maximum-allocation-vcores: The minimum allocation for every container request at the ResourceManager, in terms of virtual CPU cores. Requests smaller than this allocation will not take effect, and the specified value will be allocated the minimum number of cores. The default is 1 core.

- yarn.scheduler.maximum-allocation-vcores: The maximum allocation for every container request at the ResourceManager, in terms of virtual CPU cores. Request larger than this allocation will not take effect, and the number of cores will be capped at this value. The default is 32.
- yarn.nodemanager.resource.cpu-vcores: The number of CPU cores that can be allocated for containers.

Setting MapReduce Properties

MapReduce runs as a YARN application. Consequently, it may be necessary to adjust some of the mapred-site.xml properties as they relate to the map and reduce containers. The following properties are used to set some Java arguments and memory size for both the map and reduce containers:

- mapred.child.Java.opts provides a larger or smaller heap size for child JVMs of maps (e.g., --Xmx2048m).
- mapreduce.map.memory.mb provides a larger or smaller resource limit for maps (default = 1536MB).
- mapreduce.reduce.memory.mb provides a larger heap size for child JVMs of maps (default = 3072MB).
- mapreduce.reduce.java.opts provides a larger or smaller heap size for childreducers.

Module -3

5. a. Why should organization invest in business intelligence(BI) solutions? Are BI tools more important than IT security solutions? (08 Marks)

Ans. Business intelligence (BI) is an umbrella term that includes a variety of IT applications that are used to analyze an organization's data and communicate the information to relevant users. Business Intelligence, BI is a concept that usually involves the delivery and integration of relevant and useful business information in an organization. Its major components are data warehousing, data mining, querying, and reporting (Figure 5.1).

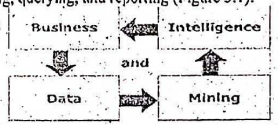


Figure 5.1 Business intelligence and data mining cycle

The nature of life and businesses is to grow. Information is the lifeblood of business. Businesses use many techniques for understanding their environment and predicting the future for their own benefit and growth. Decisions are made from facts and feelings. Data-based decisions are more effective than those based on feelings alone. Actions based on accurate data, information, knowledge, experimentation, and testing, using fresh insights, can more likely succeed and lead to sustained growth.

The organization should invest in business intelligence(BI) solutions :

Companies use BI to detect significant events and identify/monitor business trends in order to adapt quickly to their changing environment and a scenario. If effective business intelligence training is used in the organization, both decision making processes at all levels of management and tactical strategic management processes can be improved .

BI for Better Decisions: There are two main kinds of decisions:

- Strategic decisions and
- Operational decisions.

BI can help make both better.

Strategic decisions are those that impact the direction of the company. The decision to reach out to a new customer set would be a strategic decision.

Operational decisions are more routine and tactical decisions, focused on developing greater efficiency. Updating an old website with new features will be an operational decision.

In strategic decision-making, the goal itself may or may not be clear, and the same is true for the path to reach the goal.

The consequences of the decision would be apparent some time later. Thus, one is constantly scanning for new possibilities and new paths to achieve the goals. BI can help with what-if analysis of many possible scenarios. BI can also help create new ideas based on new patterns found from data mining.

Operational decisions can be made more efficient using an analysis of past data. A classification system can be created and modeled using the data of past instances to develop a good model of the domain. This model can help improve operational decisions in the future. BI can help automate operations level decision-making and improve efficiency by making millions of micro level operational decisions in a model-driven way.

For example, a bank might want to make decisions about making financial loans in a more scientific way using data-based models.

A decision-tree-based model could provide a consistently accurate loan decisions.

Developing such decision tree models is one of the main applications of data mining techniques. Effective BI has an evolutionary component, as business models evolve. When people and organizations act, new facts (data) are generated. Current business models can be tested against the new data, and it is possible that those models will not hold up well. In that case, decision models should be revised and new insights should be incorporated. An unending process of generating fresh new insights in real time can help make better decisions, and thus can be a significant competitive advantage.

BI tools more important than IT security solutions:

BI tools include data warehousing, online analytical processing, social media analytics, reporting, dashboards, querying, and data mining.

BI tools can range from very simple tools that could be considered end-user tools, to very sophisticated tools that offer a very broad and complex set of functionality. Thus, even executives can be their own BI experts, or they can rely on BI specialists to set up the BI mechanisms for them. Thus, large organizations invest in expensive sophisticated BI solutions that provide good information in real time.

A spreadsheet tool, such as Microsoft Excel, can act as an easy but effective BI tool by itself. Data can be downloaded and stored in the spreadsheet, then analyzed to produce insights, then presented in the form of graphs and tables. This system offers limited automation using macros and other features. The analytical features include basic statistical and financial functions. Pivot tables help do sophisticated what-if analysis. Add-on modules can be installed to enable moderately sophisticated statistical analysis.

A dashboarding system, such as Tableau, can offer a sophisticated set of tools for gathering, analyzing, and presenting data. At the user end, modular dashboards can be designed and redesigned easily with a graphical user interface. The back-end data analytical capabilities include many statistical functions. The dashboards are linked to data warehouses at the back end to ensure that the tables and graphs and other elements of the dashboard are updated in real time.

Data mining systems, such as IBM SPSS Modeler, are industrial strength systems that

providecapabilities to apply a wide range of analytical models on large data sets. Open source systems, such as Weka, are popular platforms designed to help mine large amounts of data to discover patterns.

b. What is the purpose of a data warehouse? (04 Marks)

Ans. Purpose of Data Warehouse lies somewhere in its definition itself i.e.a database created by combining data that is gathered through various sources that can be of different types and formats (e.g. text, sql, xml etc).

Now what you will do after storing such huge amount of data from different sources into a single database, you will analyse the data which you have accumulated and try to answer queries which were not possible or were performance intensive earlier.

In a nutshell Data warehouse is a process of collecting data , transforming it, loading into single database and then using a BI (Business Intelligence) tool to answer your analytical queries and prediction of any further questions that may arise are helpful to your domain or business.

Below are few reasons :

Improving Visibility of Data : An organization registers data in different systems, which support the various business processes. In order to create an overall picture of business operations, customers and suppliers – thus creating a single version of the truth – the data must come together in one place and made compatible. Both external (from the environment) and internal data (from ERP, CRM and financial systems) should merge into the data warehouse and then be grouped. Therefore having a single source to answers all your queries.

Improved Performance : One could use an already existing operational database if there is only single destination(Database) of all the data, yet there few constraints like performance which degrade for both operational processes and reporting processes. Therefore we create a database tuned and optimized database which will be ready to answer queries which require to bring huge amount of data and analysis.

Increase Data Quality : Stakeholders and users frequently overestimate the quality of data in the source systems. Unfortunately, source systems quite often contain data of poor quality. When we use a data warehouse, we can greatly improve the data quality, either through – were possible – correcting the data whilst loading or by tackling the problem at its source.

Faster and More advanced Reporting: The structure of both data warehouses enables end users to report in a flexible manner and to quickly perform interactive analysis on the basis of various predefined angles. They may, for example, with a single mouse click jump from year level – to quarter – to month level and quickly switch between the customer data and the product data whereby the indicator remains fixed. In this way, end users can actually juggle with the data and thus quickly gain knowledge about business operations and KPIs (Key Performance Indicator).

A data warehouse (DW) is an organized collection of integrated, subject oriented databases designed to support decision support functions. DW is organized at the right level of granularity to provide clean enterprise wide data in a standardized format for reports, queries, and analysis. DW is physically and functionally separate from an operational and transactional database. Creating a DW for analysis and queries represents significant investment in time and effort. It has to be constantly kept up-to-date for it to be useful. DW offers many business and technical benefits.

DW supports business reporting and data mining activities. It can facilitate distributed access to up-to-date business knowledge for departments and functions, thus improving business efficiency and customer service. DW can present a competitive advantage by facilitating

decision making and helping reform business processes.

DW enables a consolidated view of corporate data, all cleaned and organized.

Thus, the entire organization can see an integrated view of itself.

DW thus provides better and timely information. It simplifies data access and allows end users to perform extensive analysis. It enhances overall IT performance by not burdening the operational databases used by Enterprise Resource Planning (ERP) and other systems.

c. **Businesses need a "two-second advantage" to succeed. What does that mean to you? (04 Marks)**

Ans. Some of the examples cited for "Two Second Advantage" are:

The airlines have all the data about your bags. Why is then that you have to wait for eternity until all the bags have arrived at the baggage carousel to discover that your bag is missing and then report it to their customer service? Why can't airlines be proactive and let passengers know upfront that their bags will be arriving later?

Power companies have the data at hand on grid failures. Why do they only respond several hours after dozens of customers call and complain? Wouldn't it be better if they use the data ahead of time to prevent failures in the first place?

The Fed has all the data to take fiscal, economic and monetary decisions in real time. Why is still clinging on to an obsolete model of meeting only a few times a year to review the data and adjust the policies and rates in hindsight? Why can't the Fed be replaced by a much smarter, real time computer algorithm?

Tibco's products bring that valuable two second advantage to the enterprise for structured data. The most common form of structured data is a database, where data is stored in rows and columns.

Documents, on the other hand, represent the world of unstructured data. There is a wealth of information contained in Enterprise documents. However, this information cannot be analyzed easily since the content is not organized in a structured way. Imagine the potential an Enterprise could unleash if it were able to analyze the information scattered across thousands of documents to obtain the two second advantage.

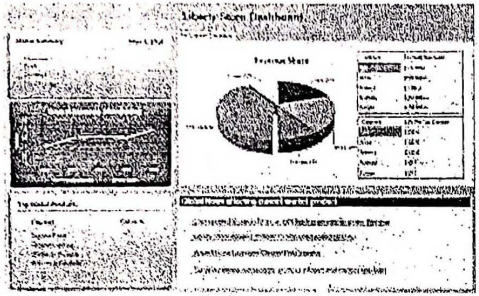
Infinote brings structure to the unstructured world of documents. It provides a platform to build tools that can give corporations the ability to extract information from their documents and become proactive instead of being reactive. In my future columns, I will explain how Infinote can help your corporation get the two second advantage by mining information from your documents.

Liberty Stores Case Exercise:

Liberty Stores Inc is a specialized global retail chain that sells organic food, organic clothing, wellness products, and education products to enlightened LOHAS (Lifestyles of the Healthy and Sustainable) citizens worldwide.

The company is 20 years old and is growing rapidly. It now operates in 5 continents, 50 countries, 150 cities, and has 500 stores. It sells 20,000 products and has 10,000 employees. The company has revenues of over $5 billion and has a profit of about 5 percent of revenue. The company pays special attention to the conditions under which the products are grown and produced. It donates about one-fifth (20 percent) of its pretax profits from global local charitable causes.

1. Create a comprehensive dashboard for the CEO of the company.
2. Create another dashboard for a country head.



OR

6. a. **What is data mining? What are supervised and unsupervised learning techniques?** **(08 Marks)**

Ans. Data mining is the art and science of discovering knowledge, insights, and patterns in data. It is the act of extracting useful patterns from an organized collection of data. Patterns must be valid, novel, potentially useful, and understandable. The implicit assumption is that data about the past can reveal patterns of activity that can be projected into the future.

Data mining is a multidisciplinary field that borrows techniques from a variety of fields. It utilizes the knowledge of data quality and data organizing from the databases area. It draws modeling and analytical techniques from statistics and computer science (artificial intelligence) areas. It also draws the knowledge of decision-making from the field of business management.

The field of data mining emerged in the context of pattern recognition in defense, such as identifying a friend-or-foe on a battlefield. Like

many other defense-inspired technologies, it has evolved to help gain a competitive advantage in business.

For example, "customers who buy cheese and milk also buy bread 90 percent of the time" would be a useful pattern for a grocery store, which can then stock the products appropriately. Similarly, "people with blood pressure greater than 160 and an age greater than 65 were at a high risk of dying from a heart stroke" is of great diagnostic value for doctors, who can then focus on treating such patients with urgent care and great sensitivity.

Past data can be of predictive value in many complex situations, especially where the pattern may not be so easily visible without the modeling technique. Here is a dramatic case of a data-driven decision-making system that beats the best of human experts. Using past data, a decision tree model was developed to predict votes for Justice Sandra Day O'Connor, who had a swing vote in a 5–4 divided US Supreme Court. All her previous decisions were coded on a few variables. What emerged from data mining was a simple four-step decision tree that was able to accurately predict her votes 71 percent of the time. In contrast, the legal analysts could at best predict correctly 59 percent of the time. (Source: Martin et al. 2004)

Supervised Learning: Classification	Machine Learning Techniques	Decision Trees
		Artificial Neural Networks
	Statistical Techniques	Regression
Unsupervised Learning: Exploration	Machine Learning Techniques	Cluster Analysis
		Association Rule Mining

Figure 6.1 Important data mining techniques

Data may be mined to help make more efficient decisions in the future. Or it may be used to explore the data to find interesting associative patterns. The right technique depends upon the kind of problem being solved (Figure 6.1).

The most important class of problems solved using data mining are classification problems. These are problems where data from past decisions is mined to extract the few rules and patterns that would improve the accuracy of the decision-making process in the future. The data of past decisions is organized and mined for decision rules or equations, which are then codified to produce more accurate decisions. Classification techniques are called supervised learning as there is a way to supervise whether the model's prediction is right or wrong.

A decision tree is a hierarchically organized branched, structured to help make decision in an easy and logical manner. Decision trees are the most popular data mining technique, for many reasons.

1. Decision trees are easy to understand and easy to use, by analysts as well as executives. They also show a high predictive accuracy.
2. They select the most relevant variables automatically out of all the available variables for decision-making.
3. Decision trees are tolerant of data quality issues and do not require much data preparation from the users.
4. Even nonlinear relationships can be handled well by decision trees.

There are many algorithms to implement decision trees. Some of the popular ones are C5, CART, and CHAID.

Regression is a relatively simple and the most popular statistical data mining technique. The goal is to fit a smooth well-defined curve to the data. Regression analysis techniques, for example, can be used to model and predict the energy consumption as a function of daily temperature. Simply plotting the data shows a nonlinear curve. Applying a nonlinear regression equation will fit the data very well with high accuracy. Thus, the energy consumption on any future day can be predicted using this equation.

Artificial neural network (ANN) is a sophisticated data mining technique from the Artificial Intelligence stream in Computer Science. It mimics the behavior of human neural structure. Neurons receive stimuli, process them, and communicate their results to other neurons successively, and eventually a neuron outputs a decision. A decision task may be processed by just one neuron and the result may be communicated soon. Alternatively, there could be many layers of neurons involved in a decision task, depending upon the complexity of the domain. The neural network can be trained by making a decision over and over again with many data points. It will continue to learn by adjusting its internal computation and communication parameters based on feedback received on its previous decisions. The intermediate values

passed within the layers of neurons may not make intuitive sense to an observer. Thus, the neural networks are considered a black-box system.

At some point, the neural network will have learned enough and begin to match the predictive accuracy of a human expert or alternative classification techniques. The predictions of some ANNs that have been trained over a long period of time with a large amount of data have become decisively more accurate than human experts. At that point, the ANNs can begin to be seriously considered for deployment, in real situations in real time.

ANNs are popular because they are eventually able to reach a high predictive accuracy. ANNs are also relatively simple to implement and do not have any issues with data quality. ANNs require a lot of data to train to develop good predictive ability.

Cluster analysis is an exploratory learning technique that helps in identifying a set of similar groups in the data. It is a technique used for automatic identification of natural groupings of things. Data instances that are similar to (or near) each other are categorized into one cluster, while data instances that are very different (or far away) from each other are categorized into separate clusters. There can be any number of clusters that could be produced by the data. The K-means technique is a popular technique and allows the user guidance in selecting the right number (K) of clusters from the data.

Clustering is also known as the segmentation technique. The technique shows the clusters of things from past data. The output is the centroids for each cluster and the allocation of data points to their cluster.

The centroid definition is used to assign new data instances that can be assigned to their cluster homes. Clustering is also a part of the artificial intelligence family of techniques.

Association rules are a popular data mining method in business, especially where selling is involved. Also known as market basket analysis, it helps in answering questions about cross-selling opportunities. This is the heart of the personalization engine used by e-commerce sites like Amazon.com and streaming movie sites like Netflix.com. The technique helps find interesting relationships (affinities) between variables (items or events). These are represented as rules of the form $X \Rightarrow Y$, where X and Y are sets of data items. A form of unsupervised learning, it has no dependent variable; and there are no right or wrong answers. There are just stronger and weaker affinities. Thus, each rule has a confidence level assigned to it. A part of the machine-learning family, this technique achieved legendary status when a fascinating relationship was found in the sales of diapers and beers.

b. Why is data preparation so important and time consuming? (04 Marks)

Ans. Data cleansing and preparation is a laborintensive or semiautomated activity that can take up to 60 to 70 percent of the time needed for a data mining project.

1. Duplicate data needs to be removed. The same data may be received from multiple sources. When merging the data sets, data must be de-duped.
2. Missing values need to be filled in, or those rows should be removed from analysis. Missing values can be filled in with average or modal or default values.
3. Data elements may need to be transformed from one unit to another. For example, total costs of health care and the total number of patients may need to be reduced to cost/patient to allow comparability of that value.
4. Continuous values may need to be binned into a few buckets to help with some analyses. For example, work experience could be binned as low, medium, and high.
5. Data elements may need to be adjusted to make them comparable over time. For example, currency values may need to be adjusted for inflation; they would need to be converted to the same base year for comparability. They may need to be converted to a common currency.

6. Outlier data elements need to be removed after careful review, to avoid the skewing of results. For example, one big donor could skew the analysis of alumni donors in an educational setting.
7. Any biases in the selection of data should be corrected to ensure the data is representative of the phenomena under analysis. If the data includes many more members of one gender than is typical of the population of interest, then adjustments need to be applied to the data.
8. Data should be brought to the same granularity to ensure comparability. Sales data may be available daily, but the sales person compensation data may only be available monthly. To relate these variables, the data must be brought to the lowest common denominator, in this case, monthly.
9. Data may need to be selected to increase information density. Some data may not show much variability, because it was not properly recorded or for any other reasons. This data may dull the effects of other differences in the data and should be removed to improve the information density of the data.

c. What is data visualization? How would you judge the quality of data visualizations? (04 Marks)

Ans. Data Visualization is the art and science of making data easy to understand and consume, for the end user. Ideal visualization shows the right amount of data, in the right order, in the right visual form, to convey the high priority information. The right visualization requires an understanding of the consumer's needs, nature of the data, and the many tools and techniques available to present data. The right visualization arises from a complete understanding of the totality of the situation. One should use visuals to tell a true, complete and fast-paced story.

Data visualization is the last step in the data life cycle. This is where the data is processed for presentation in an easy-to-consume manner to the right audience for the right purpose. The data should be converted into a language and format that is best preferred and understood by the consumer of data. The presentation should aim to highlight the insights from the data in an actionable manner. If the data is presented in too much detail, then the consumer of that data might lose interest and the insight.

The quality of data visualizations can be judged by Excellence Visualization.

Data can be presented in the form of rectangular tables, or it can be presented in colorful graphs of various types. "Small, non comparative, highly-labeled data sets usually belong in tables". However, as the amount of data grows, graphs are preferable. Graphics help give shape to data. Tufte, a pioneering expert on data visualization, presents the following objectives for graphical excellence:

1. Show, and even reveal, the data: The data should tell a story, especially a story hidden in large masses of data. However, reveal the data in context, so the story is correctly told.
2. Induce the viewer to think of the substance of the data: The format of the graph should be so natural to the data, that it hides itself and lets data shine.
3. Avoid distorting what the data have to say: Statistics can be used to lie. In the name of simplifying, some crucial context could be removed leading to distorted communication.
4. Make large data sets coherent: By giving shape to data, visualizations can help bring the data together to tell a comprehensive story.
5. Encourage the eyes to compare different pieces of data: Organize the chart in ways the eyes would naturally move to derive insights from the graph.
6. Reveal the data at several levels of detail: Graphs leads to insights, which raise further curiosity, and thus presentations should help get to the root cause.

7. Serve a reasonably clear purpose – informing or decision-making.
8. Closely integrate with the statistical and verbal descriptions of the dataset: There should be no separation of charts and text in presentation. Each mode should tell a complete story. Intersperse text with the map/graphic to highlight the main insights.

Module -4

7. a. What is a decision tree? Why are decision trees the most popular classification technique? (02 Marks)

Ans. A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome(categorical or continues value).The whole idea is to create a tree like this for the entire data and process a single outcome at every leaf(or minimize the error in every leaf)

Decision trees are a simple way to guide one's path to a decision. The decision may be a simple binary one, whether to approve a loan or not. Or it may be a complex multi-valued decision, as to what may be the diagnosis for a particular sickness. Decision trees are hierarchically branched structures that help one come to a decision based on asking certain questions in a particular sequence.

Decision trees are one of the most widely used techniques for classification. A good decision tree should be short and ask only a few meaningful questions. They are very efficient to use, easy to explain, and their classification accuracy is competitive with other methods. Decision trees can generate knowledge from a few test instances that can then be applied to a broad population. Decision trees are used mostly to answer relatively simple binary decisions.

b. What is a regression model? What is a scatter plot? How does it help?(6 Marks)

Ans. Regression is a well-known statistical technique to model the predictive relationship between several independent variables (DVs) and one dependent variable. The objective is to find the best-fitting curve for a dependent variable in a multidimensional space, with each independent variable being a dimension. The curve could be a straight line, or it could be a nonlinear curve.

The quality of fit of the curve to the data can be measured by a coefficient of correlation (r), which is the square root of the amount of variance explained by the curve.

The key steps for regression are simple:

1. List all the variables available for making the model.
2. Establish a Dependent Variable (DV) of interest.
3. Examine visual (if possible) relationships between variables of interest.
4. Find a way to predict DV using the other variables.

A scatter plot (or scatter diagram) is a simple exercise for plotting all data points between two variables on a two-dimensional graph. It provides a visual layout of where all the data points are placed in that two-dimensional space.

The scatter plot can be useful for graphically intuiting the relationship between two variables. Here is a picture (Figure 7.1) that shows many possible patterns in scatter diagrams.

Figure.7.1: Scatter plots showing types of relationships among two variables

(Source: Groebner et al. 2013)

Chart (a) shows a very strong linear relationship between the variables x and y. That means the value of y increases proportionally with x. Chart (b) also shows a strong linear relationship between the variables x and y. Here it is an inverse relationship. That means the value of y decreases proportionally with x.

Chart (c) shows a curvilinear relationship. It is an inverse relationship, which means that the value of y decreases proportionally with x. However, it seems a relatively well-defined relationship, like an arc of a circle, which can be represented by a simple quadratic equation (quadratic means the power of two, that is, using terms like x2 and y2). Chart (d) shows a positive curvilinear relationship. However, it does not seem to resemble a regular shape, and thus would not be a strong relationship. Charts (e) and (f) show no relationship.

That means variables x and y are independent of each other. Charts (a) and (b) are good candidates that model a simple linear regression model (the terms regression model and regression equation can be used interchangeably). Chart (c) too could be modeled with a little more complex, quadratic regression equation. Chart (d) might require an even higher order polynomial regression equation to represent the data.

Charts (e) and (f) have no relationship, thus, they cannot be modeled together, by regression or using any other modeling tool.

c. Examine the steps in developing a neural network for predicting stock prices. What kind of objective function and what kind of data would be required for a good stock price predictor system using ANN? (08 Marks)

Ans. It takes resources, training data, and skill and time to develop a neural network. Most data mining platforms offer at least the MLP algorithm to implement a neural network. The steps required to build an ANN are as follows:

1. Gather data: Divide into training data and test data. The training data needs to be further divided into training data and validation data.
2. Select the network architecture, such as feedforward network.
3. Select the algorithm, such as Multilayer Perception.
4. Set network parameters.
5. Train the ANN with training data.
6. Validate the model with validation data.
7. Freeze the weights and other parameters.
8. Test the trained network with test data.
9. Deploy the ANN when it achieves good predictive accuracy.

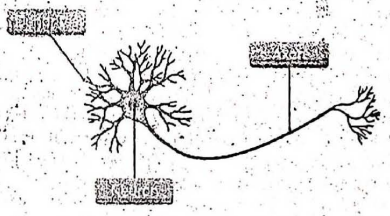
Other neural network architectures include probabilistic networks and self-organizing feature

maps.

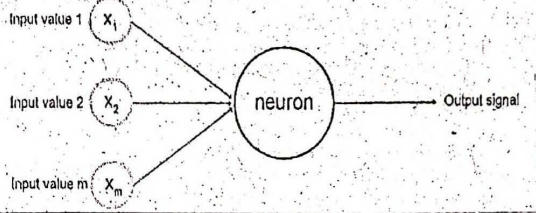
Training an ANN: Training data is split into three parts	
Training set	This data set is used to adjust the "weights on the neural network (~ 60%).
Validation set	This data set is used to minimize over fitting and verifying accuracy (~ 2X?%).
Testing set	This data set is used only for testing the final solution in order to confirm the actual predictive power of the network (~ 20 %)
k - fold cross - validation	approach means that the data is divided into k equal pieces, and the learning process is repeated k-times with each pieces becoming the training set. This process leads to less bias and more accuracy, but is more time consuming.

Machine learning has proved to improve efficiencies significantly, and there are many jobs which have been replaced by smarter and faster machines using artificial intelligence or machine learning. The stock markets are no exceptions to this. Today, there are several Machine Learning algorithms running in the live markets. These algorithms often provide greater returns than other alternate algorithms or sometimes even higher than experienced traders. In this article, I will talk about the concepts involved in a neural network and how it can be applied to predict stock prices in the live markets. Let us start by understanding what a neuron is.

Neuron



This is the neuron that you must be familiar with, well if you aren't you should now be grateful that you can understand this because there are billions of neurons in your brain. There are three components to a neuron, the dendrites, the axon and the main body of the neuron. The dendrites are the receivers of the signal and the axon is the transmitter. Alone, a neuron is not of much use, but when it is connected to other neurons, it does several complicated computations and helps operate the most complicated machine on our planet, the human body.

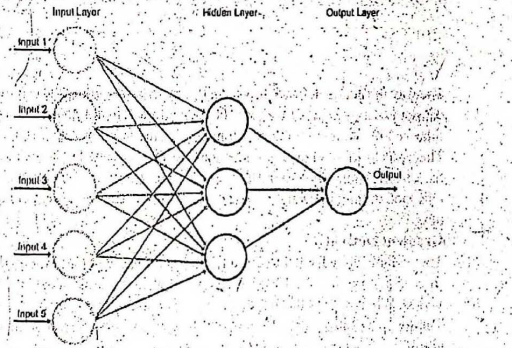


A computer neuron is built in a similar manner, as shown in the diagram. There are inputs to the neuron marked with yellow circles, and the neuron emits an output signal after some computation. The input layer resembles the dendrites of the neuron and the output signal is the axon. Each input signal is assigned a weight, w_i. This weight is multiplied by the input value and the neuron stores the weighted sum of all the input variables. These weights are computed in the training phase of the neural network through concepts called gradient descent and back propagation, we will cover these topics in our subsequent blog posts on Neural Networks. An activation function is then applied to the weighted sum, which results in the output signal of the neuron. The input signals are generated by other neurons, i.e, the output of other neurons, and the network is built to make predictions/computations in this manner. This is the basic idea of a neural network. We will look at each of these concepts in more detail in this article.

Working of Neural Networks

We will look at an example to understand the working of neural networks. The input layer consists of the parameters that will help us arrive at an output value or make a prediction. Our brains essentially have five basic input parameters, which are our senses to touch, hear, see, smell and taste. The neurons in our brain create more complicated parameters such as emotions and feelings, from these basic input parameters. And our emotions and feelings, make us act or take decisions which is basically the output of the neural network of our brains. Therefore, there are two layers of computations in this case before making a decision. The first layer takes in the five senses as inputs and results in emotions and feelings, which are the inputs to the next layer of computations, where the output is a decision or an action. Hence, in this extremely simplistic model of the working of the human brain, we have one input layer, two hidden layers, and one output layer. Of course from our experiences, we all know that the brain is much more complicated than this, but essentially this is how the computations are done in our brain.

To understand the working of a neural network, let us consider a simple stock price prediction example, where the OHLCV (Open-High-Low-Close-Volume) values are the input parameters, there is one hidden layer and the output consists of the prediction of the stock price.



There are five input parameters as shown in the diagram, the hidden layer consists of 3 neurons and the resultant in the output layer is the prediction for the stock price. The 3 neurons in the hidden layer will have different weights for each of the five input parameters and might have different activation functions, which will activate the input parameters according to various combinations of the inputs. For example, the first neuron might be looking at the volume and the difference between the Close and the Open price and might be ignoring the High and Low prices. In this case, the weights for High and Low prices will be zero. Based on the weights that the model has trained itself to attain, an activation function will be applied to the weighted sum in the neuron, this will result in an output value for that particular neuron. Similarly, the other two neurons will result in an output value based on their individual activation functions and weights. Finally, the output value or the predicted value of the stock price will be the sum of the three output values of each neuron. This is how the neural network will work to predict stock prices.

Conclusion

There are two ways to code a program for performing a specific task. One is to define all the rules required by the program to compute the result given some input to the program. The other way is to develop the framework upon which the code will learn to perform the specific task by training itself on a dataset through adjusting the result it computes to be as close to the actual results which have been observed. This process is called training the model, we will now look at how our neural network will train itself to predict stock prices.

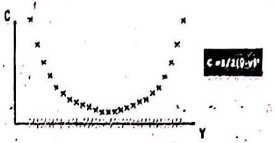
The neural network will be given the dataset, which consists of the OHLCV data as the input and as the output, we would also give the model the Close price of the next day, this is the value that we want our model to learn to predict. The actual value of the output will be represented by 'y' and the predicted value will be represented by y^, y hat. The training of the model involves adjusting the weights of the variables for all the different neurons present in the neural network. This is done by minimizing the 'Cost Function'. The cost function, as the name suggests is the cost of making a prediction using the neural network. It is a measure of how far off the predicted value, y^, is from the actual or observed value, y. There are many cost functions that are used in practice, the most popular one is computed as half of the sum of squared differences of the actual and predicted values for the training dataset.

$$C = \sum_{i=1}^{n} 1/2\,(\hat{y} - y)^2$$

The way the neural network trains itself is by first computing the cost function for the training dataset for a given set of weights for the neurons. Then it goes back and adjusts the weights, followed by computing the cost function for the training dataset based on the new weights. The process of sending the errors back to the network for adjusting the weights is called backpropagation. This is repeated several times till the cost function has been minimized. We will look at how the weights are adjusted and the cost function is minimized in more detail next.

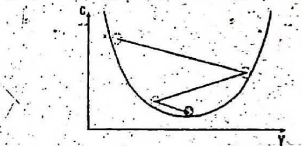
Gradient Descent

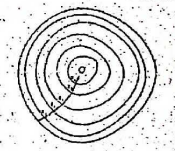
The weights are adjusted to minimize the cost function. One way to do this is through brute force. Suppose we take 1000 values for the weights, and evaluate the cost function for these values. When we plot the graph of the cost function, we will arrive at a graph as shown below. The best value for weights would be the cost function corresponding to the minima of this graph.



This approach could be successful for a neural network involving a single weight which needs to be optimized. However, as the number of weights to be adjusted and the number of hidden layers increases, the number of computations required will increase drastically. The time it will require to train such a model will be extremely large even on the world's fastest supercomputer. For this reason, it is essential to develop a better, faster methodology for computing the weights of the neural network. This process is called Gradient Descent.

Gradient descent involves analyzing the slope of the curve of the cost function. Based on the slope we adjust the weights, to minimize the cost function in steps rather than computing the values for all possible combinations. The visualization of Gradient descent is shown in the diagrams below. The first plot is a single value of weights and hence is two dimensional. It can be seen that the red ball moves in a zig-zag pattern to arrive at the minimum of the cost function. In the second diagram, we have to adjust two weights in order to minimize the cost function. Therefore, we can visualize it as a contour, as shown in the graph, where we are moving in the direction of the steepest slope, in order to reach the minima in the shortest duration. With this approach, we do not have to do many computations and as a result, the computations do not take very long, making the training of the model a feasible task.





Gradient descent can be done in three possible ways, batch gradient descent, stochastic gradient descent and mini-batch gradient descent. In batch gradient descent, the cost function is computed by summing all the individual cost functions in the training dataset and then computing the slope and adjusting the weights. In stochastic gradient descent, the slope of the cost function and the adjustments of weights are done after each data entry in the training dataset. This is extremely useful to avoid getting stuck at a local minima if the curve of the cost function is not strictly convex. Each time you run the stochastic gradient descent, the process to arrive at the global minima will be different. Batch gradient descent may result in getting stuck with a suboptimal result if it stops at local minima. The third type is the mini-

batch gradient descent, which is a combination of the batch and stochastic methods. Here, we create different batches by clubbing together multiple data entries in one batch. This essentially results in implementing the stochastic gradient descent on bigger batches of data entries in the training dataset. Next, let us understand how backpropagation works to adjust the weights according to the error which had been generated.

Backpropagation

Backpropagation is an advanced algorithm which enables us to update all the weights in the neural network simultaneously. This drastically reduces the complexity of the process to adjust weights. If we were not using this algorithm, we would have to adjust each weight individually by figuring out what impact that particular weight has on the error in the prediction. Let us look at the steps involved in training the neural network with Stochastic Gradient Descent:

- Initialize the weights to small numbers very close to 0 (but not 0)
- Forward propagation – the neurons are activated from left to right, by using the first data entry in our training dataset, until we arrive at the predicted result y
- Measure the error which will be generated
- Backpropagation – the error generated will be back propagated from right to left, and the weights will be adjusted according to the learning rate
- Repeat the previous three steps, forward propagation, error computation and back propagation on the entire training dataset
- This would mark the end of the first epoch, the successive epochs will begin with the weight values of the previous epochs, we can stop this process when the cost function converges within a certain acceptable limit

OR

8. a. What is a neural network? How does it work? Explain the Design Principles of an Artificial Neural Network. (08 Marks)

Ans. Artificial Neural Networks (ANN) are inspired by the information processing model of the mind/brain. The human brain consists of billions of neurons that link with one another in an intricate pattern. Every neuron receives information from many other neurons, processes it, gets excited or not, and passes its state information to other neurons.

Just like the brain is a multipurpose system, so also the ANNs are very versatile systems. They can be used for many kinds of pattern recognition and prediction. They are also used for classification, regression, clustering, association, and optimization activities. They are used in finance, marketing, manufacturing, operations, information systems applications, and so on.

ANNs are composed of a large number of highly interconnected processing elements (neurons) working in a multi-layered structures that receive inputs, process the inputs, and produce an output. An ANN is designed for a specific application, such as pattern recognition or data classification, and trained through a learning process. Just like in biological systems, ANNs make adjustments to the synaptic connections with each learning instance.

ANNs are like a black box trained into solving a particular type of problem, and they can develop high predictive powers. Their intermediate synaptic parameter values evolve as the system obtains feedback on its predictions, and thus an ANN learns from more training data (Figure 8.1).

Figure 8.1 General ANN model

The Design Principl of ANN :

1. A neuron is the basic processing unit of the network. The neuron (or processing element) receives inputs from its preceding neurons (or PEs), does some nonlinear weighted computation on the basis of those inputs, transforms the result into its output value, and then passes on the output to the next neuron in the network (Figure 8.2). X's are the inputs, w's are the weights for each input, and y is the output.

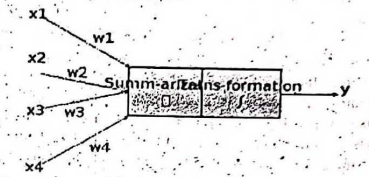


Figure 8.2: Model for a single artificial neuron

2. A Neural network is a multi-layered model. There is at least one input neuron, one output neuron, and at least one processing neuron. An ANN with just this basic structure would be a simple, single-stage computational unit. A simple task may be processed by just that one neuron and the result may be communicated soon. ANNs however, may have multiple layers of processing elements in sequence. There could be many neurons involved in a sequence depending upon the complexity of the predictive action. The layers of PEs could work in sequence, or they could work in parallel (Figure 8.3).

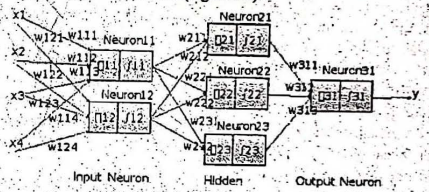


Figure 8.3: Model for a multi-layer ANN

3. The processing logic of each neuron may assign different weights to the various incoming input streams. The processing logic may also use nonlinear transformation, such as a sigmoid function, from the processed values to the output value. This processing logic and the intermediate weight and processing functions are just what works for the system as a whole, in its objective of solving a problem collectively. Thus, neural networks are considered to be an opaque and a black-box system.

4. The neural network can be trained by making similar decisions over and over again with many training cases. It will continue to learn by adjusting its internal computation and communication based on feedback about its previous decisions. Thus, the neural networks become better at making a decision as they handle more and more decisions.

Depending upon the nature of the problem and the availability of good training data, at some point the neural network will learn enough and begin to match the predictive accuracy of a human expert. In many practical situations, the predictions of ANN, trained over a long period of time with a large number of training data, have begun to decisively become more accurate than human experts. At that point ANN can begin to be seriously considered for deployment in real situations in real time.

b. What is unsupervised learning? When is it used? (04 Marks)

Ans. Unsupervised learning, by contrast, does not begin with a target variable. Instead the objective is to find groups of similar records in the data. One can think of unsupervised learning as a form of data compression: we search for a moderate number of representative records to summarize or stand in for the original database. Consider a mobile telecommunications company with 20 million customers. The company database will likely contain various categories of information including customer characteristics such as age and postal code, product information describing the customer's mobile handset, features of the plans the subscriber has selected, details of the subscribers use of plan features, and billing and payment information. Although it is almost certain that no two subscribers will be identical on every detail in their customer records, we would expect to find groups of customers that are very similar in their overall pattern of demographics, selected equipment, plan use, and spending and payment behavior. If we could find say 30 representative customer types such that the bulk of customers are well described as belonging to their "type", this information could be very useful for marketing, planning, and new product development. We cannot promise that we can find clusters or groupings in data that you will find useful. But we include a method quite distinct from that found in other statistical or data mining software. CART and other Salford data mining modules now include an approach to cluster analysis, density estimation and unsupervised learning using ideas that we trace to Leo Breiman, but which may have been known informally in among statisticians at Stanford and elsewhere for some time. The method detects structure in data by contrasting original data with randomized variants of that data. Analysts use this method implicitly when viewing data graphically to identify clusters or other structure in data visually. Take for example customer ages and handsets owned. If there is a pattern in the data then we expect to see certain handsets owned by people in their early 20's, and rather different handsets owned by customers in their early 30's. If every handset is just as likely to be owned in every age group then there is no structure relating these two data dimensions. The method we use generalizes this everyday detection idea to high dimensions.

The method consists of these steps:

Make a copy of the original data, and then randomly scramble each column of data separately. As an example, starting with data typical of a mobile phone company, suppose we randomly exchanged date of birth information at random in our copy of the database. Each customer record would likely contain age information belonging to another customer. We now repeat this process in every column of the data. Breiman uses a variant in which each column of original data is replaced with a bootstrap resample of the column and you can use either method in Salford software.

Note that all we have done is moved information about in the database, but other than moving data we not changed anything. So aggregates such as averages and totals will not have changed. Any one customer record is now a "Frankenstein" record, with item of information having been obtained from a different customer. Thus, date of birth might be from customer 101135, the service plan taken from customer 456779 and the spend data from 987001.

Now append the scrambled data set to the original data. We therefore now have the same number of columns as before but twice as many rows. The top portion of the data is the original data and the bottom portion will be the scrambled copy. Add a new column to the data to label records by their data source ("Original" vs. "Copy").

Generate a predictive model to attempt to discriminate between the Original and Copy data sets. If it is impossible to tell, after the fact, which records are original and which are random artifacts then there is no structure in the data. If it is easy to tell the difference then there is strong structure in the data.

In the CART model separating the Original from the Copy records, nodes with a high fraction of Original records define regions of high density and qualify as potential "clusters". Such nodes reveal patterns of data values, which appear frequently in the real data but not in the randomized artifact.

We don not expect the optimal sized tree for cluster detection to be the most accurate separator of Original from Copy records. We recommend that you prune back to a tree size that reveals interesting data groupings.

This approach to unsupervised learning represents an important advance in clustering technology because:

- Variable selection is not necessary and different clusters may be defined on different groups of variable
- Preprocessing or rescaling of the data is unnecessary as these clustering methods are not influenced by how data is scaled
- Missing values present no challenges as the methods automatically manage missing data
- The CART-based clustering gives easy control over the number of clusters and helps select the optimal number

c. What are association rules? How do they help? (04 Marks)

Ans. Association rule mining is a popular, unsupervised learning technique, used in business to help identify shopping patterns. It is also known as market basket analysis. It helps find interesting relationships (affinities) between variables (items or events). Thus, it can help cross-sell related items and increase the size of a sale.

All data used in this technique is categorical. There is no dependent variable. It uses machine-learning algorithms. The fascinating "relationship between sales of diapers and beers" is how it is often explained in popular literature. This technique accepts as input the raw point-of-sale transaction data. The output produced is the description of the most frequent affinities among items. An example of an association rule would be, "a Customer who bought a laptop computer and virus protection software also bought an extended service plan 70 percent of the time."

In business environments a pattern or knowledge can be used for many purposes. In sales and marketing, it is used for cross-marketing and cross selling, catalog design, e-commerce site design, online advertising optimization, product pricing, and sales/promotion configurations. This analysis can suggest not to put one item on sale at a time, and instead to create a bundle of products promoted as a package to sell other nonselling items.

In retail environments, it can be used for store design. Strongly associated items can be kept close tougher for customer convenience. Or they could be placed far from each other so that the customer has to walk the aisles and by doing so is potentially exposed to other items.

In medicine, this technique can be used for relationships between symptoms and illnesses; diagnosis and patient characteristics/treatments; genes and their functions; and so on.

Representing Association Rules

A generic rule is represented between a set X and Y: X ⇒ Y [S%, C%]
X, Y: products and/or services
X: Left-hand-side (LHS or Antecedent)
Y: Right-hand-side (RHS or Consequent)
S: Support: how often X and Y go together in the total transaction set
C: Confidence: how often Y goes together with X
Example: {Laptop Computer, Antivirus Software} ⇒{Extended Service Plan} [30%, 70%]

Module -5

9. a. Why is text mining useful in the age of social media? (04 Marks)

Ans. Text mining is the art and science of discovering knowledge, insights and patterns from an organized collection of textual databases. Textual mining can help with frequency analysis of important terms, and their semantic relationships.

Text is an important part of the growing data in the world. Social media technologies have enabled users to become producers of text and images and other kinds of information. Text mining can be applied to large-scale social media data for gathering preferences, and measuring emotional sentiments. It can also be applied to societal, organizational and individual scales.

Text mining works on texts from practically any kind of sources from any business or non-business domains, in any formats including Word documents, PDF files, XML files, text messages, etc. Here are some representative examples:

1. In the legal profession, text sources would include law, court deliberations, court orders, etc.
2. In academic research, it would include texts of interviews, published research articles, etc.
3. The world of finance will include statutory reports, internal reports, CFO statements, and more.
4. In medicine, it would include medical journals, patient histories, discharge summaries, etc.
5. In marketing, it would include advertisements, customer comments, etc.
6. In the world of technology and search, it would include patent applications, the whole of information on the world-wide web, and more.

b. What is a Naive-Bayes technique? What does Naive & Bayes stand for? (08 Marks)

Ans. **Naive Bayes algorithm :** Naive Bayes is a simple technique for constructing classifiers and models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

Naive Bayes algorithm works :

Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play' (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table			
Weather	No	Yes	
Overcast		4	=4/14 = 0.29
Rainy	3	2	=5/14 = 0.36
Sunny	2	3	=5/14 = 0.36
All	5	9	
	=5/14 = 0.36	=9/14 = 0.64	

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will play if weather is sunny. Is this statement is correct?

We can solve it using above discussed method of posterior probability.

P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

Naive Bayes stand for:

The word Bayes refers to Baysian analysis (based on the work of the mathematician Thomas Bayes) which computes the probability of a new occurrence not only the recent record, but also on the basis of prior experience.

The word Naïve represents the strong assumption that all the parameters of the instances are independent variables with little or no correlation. Thus if people are identified by their height , weight, age, gender, all these variables are assumed to be independent of each other.

c. What is a support vector machine? (04 Marks)

Ans. A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

Confusing? Don't worry, we shall learn in laymen terms.

Suppose you are given plot of two label classes on graph as shown in image (A). Can you decide a separating line for the classes?

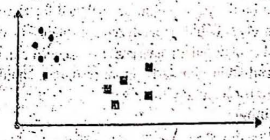


Image A: Draw a line that separates black circles and blue squares.

You might have come up with something similar to following image (image B). It fairly separates the two classes. Any point that is left of line falls into black circle class and on right

falls into blue square class. Separation of classes. That's what SVM does. It finds out a line/ hyper-plane (in multidimensional space that separate outs classes). Shortly, we shall discuss why I wrote multidimensional space.

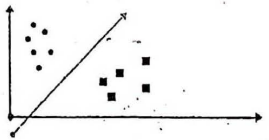


Image B: Sample cut to divide into two classes.

OR

10. a. What are the three types of web mining? (10 Marks)

Ans. The web could be analyzed for its structure as well as content. The usage pattern of web pages could also be analyzed. Depending upon objectives, web mining can be divided into three different types: Web usage mining, Web content mining and Web structure mining (Figure 10.1).

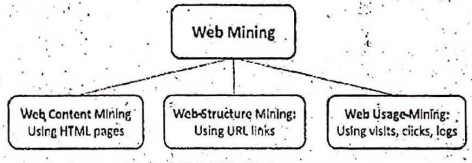


Figure: 10.1 Web Mining structure

Web content mining

A website is designed in the form of pages with a distinct URL (universal resource locator). A large website may contain thousands of pages. These pages and their content is managed using specialized software systems called Content Management Systems. Every page can have text, graphics, audio, video, forms, applications, and more kinds of content including user generated content.

The websites keep a record of all requests received for its page/URLs, including the requester information using 'cookies'. The log of these requests could be analyzed to gauge the popularity of those pages among different segments of the population. The text and application content on the pages could be analyzed for its usage by visit counts. The pages on a website themselves could be analyzed for quality of content that attracts most users. Thus the unwanted or unpopular pages could be weeded out, or they can be transformed with different content and style. Similarly, more resources could be assigned to keep the more popular pages more fresh and inviting.

Web structure mining

The Web works through a system of hyperlinks using the hypertext protocol (http). Any page can create a hyperlink to any other page, it can be linked to by another page. The intertwined or self-referral nature of web lends itself to some unique network analytical algorithms. The structure of Web pages could also be analyzed to examine the pattern of hyperlinks among

pages. There are two basic strategic models for successful websites: Hubs and Authorities.

1. Hubs: These are pages with a large number of interesting links. They serve as a hub, or a gathering point, where people visit to access a variety of information. Media sites like Yahoo.com, or government sites would serve that purpose. More focused sites like Traveladvisor.com and yelp.com could aspire to becoming hubs for new emerging areas.

2. Authorities: Ultimately, people would gravitate towards pages that provide the most complete and authoritative information on a particular subject. This could be factual information, news, advice, user reviews etc. These websites would have the most number of inbound links from other websites. Thus Mayoclinic.com would serve as an authoritative page for expert medical opinion. NYtimes.com would serve as an authoritative page for daily news.

Web usage mining

As a user clicks anywhere on a webpage or application, the action is recorded by many entities in many locations. The browser at the client machine will record the click, and the web server providing the content would also make a record of the pages served and the user activity on those pages. The entities between the client and the server, such as the router, proxy server, or ad server, too would record that click.

The goal of web usage mining is to extract useful information and patterns from data generated through Web page visits and transactions. The activity data comes from data stored in server access logs, referrer logs, agent logs, and client-side cookies. The user characteristics and usage profiles are also gathered directly, or indirectly, through syndicated data. Further, metadata, such as page attributes, content attributes, and usage data are also gathered.

The web content could be analyzed at multiple levels (Figure 10.2).

1. The server side analysis would show the relative popularity of the web pages accessed. Those websites could be hubs and authorities.

2. The client side analysis could focus on the usage pattern or the actual content consumed and created by users.

1. Usage pattern could be analyzed using 'clickstream' analysis, i.e. analyzing web activity for patterns of sequence of clicks, and the location and duration of visits on websites. Clickstream analysis can be useful for web activity analysis, software testing, market research, and analyzing employee productivity.

2. Textual information accessed on the pages retrieved by users could be analyzed using text mining techniques. The text would be gathered and structured using the bag-of-words technique to build a Term-document matrix. This matrix could then be mined using cluster analysis and association rules for patterns such as popular topics, user segmentation, and sentiment analysis.

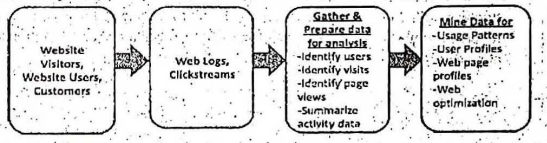


Figure: 10.2 Web Usage Mining architecture

Web usage mining has many business applications. It can help predict user behavior based on previously learned rules and users' profiles, and can help determine lifetime value of clients. It can also help design cross-marketing strategies across products, by observing association

rules among the pages on the website. Web usage can help evaluate promotional campaigns and see if the users were attracted to the website and used the pages relevant to the campaign. Web usage mining could be used to present dynamic information to users based on their interests and profiles. This includes targeted online ads and coupons at user groups based on user access patterns.

b. **What is social network analysis? Explain the applications of SNA? (06 Marks)**

Ans. Social Network Analysis (SNA) is the mapping and measuring of relationships and streams between people, groups, organizations, computers, URLs and other sources of information that are connected. Management consultants in particular use Social Network Analysis to map their business relations and further investigate their mutual relationships. This can be compared to a central nervous system that connects everything.

Social networks are a graphical representation of relationships among people and/ or entities. SNA is the art and science of discouvering patterns of interaction and influence within the participants in a network. These participants could be people, organizations, machines, concepts, or any other kind of entities. There are two major levels of social network analysis: discovering sub- networks within the network, and ranking the nodes to find more important nodes or hubs.

Computing the relative influence of each node is done on the basis of an input- output matrix of flows of influence among the nodes.

Applications of SNA:

- Self-awareness
- Communities
- Marketing
- Public health

Self –awareness: Visualizing his/her social network can help a person organize their relationships and support network

Communities: SNA can help identification, construction, and strengthening of networks within communities to build wellness, comfort, and resilience. Analysis of joint authoring relationships and citations help identify subnet works of specializations of knowledge in an academic field. Researchers at Northwestern university found that the most determinant factor in the success of a broad way play was the strength amongst the crew and cast.

Marketing: there is a popular network insight that any two people are related to each other through at most seven degrees of links. Organizations can use this insight to reach out with their message to large number of people and also to listen actively to opinion leaders as ways to understand their customers needs and behaviors. Politicians can reach out to opinion leaders to get their message out.

Public health: Awareness of network can help identify the paths that certain diseases take to spread. Public health professionals can isolate and contain diseases before they expand to other networks.

Eighth Semester B.E. Degree Examination,
CBCS - Model Question Paper - 2
BIG DATA ANALYTICS

Time: 3 hrs. Max. Marks: 80

Note : *Answer any FIVE full questions, selecting ONE full question from each module.*

Module - 1

1. a. **Explain General HDFS commands in detail (08 Marks)**

Ans. **General HDFS Commands:** The version of HDFS can be found from the version option. Examples in this section are run on the HDFS version shown here:

```
$ hdfs version
Hadoop 2.6.0.2.2.4.2-2
Subversion git@github.com:hortonworks/hadoop.git -r
22a563ebe448969d07902aed869ac13c652b2872Compiledbyjenkinson2015-03-31T19:49Z
Compiled with protoc 2.5.0
From source with checksum b3481c2cdbe2d181f2621331926e267 This command was run
using /usr/hdp/2.2.4.2-2/hadoop/hadoop-Common-2.6.0.2.2.4.2-2.jar
```

HDFS provides a series of commands similar to those found in a standard POSIX file system. A list of those commands can be obtained by issuing the following command. Several of these commands will be highlighted here under the user account hdfs.

```
$ hdfs dfs
Usage: hadoop fs [generic options] [-appendToFile <localsrc>...<dst>]
[-cat [-ignoreCrc] <src>...]
[-checksum <src>...]
[-chgrp [-R] GROUP PATH...]
[-chown [-R] <MODE[,MODE]...| OCTALMODE> PATH...]
[-copyFromLocal [-f] [-p] [-l] <localsrc>...<dst>]
[-count [-q] [-h] <path>...]
[-cp [-f] [-p] -p[topax] ] <src>...<dst>]
[-createSnapshot <snapshotDir> [<snapshotName>] ]
[-deleteSnapshot <snapshotDir> <snapshotName>]
[-df [-h] <path>...]
[-du [-s] [-h] <path>...] [-expunge]
[-get [-p] [-ignoreCrc] [-crc] <src>...<localdst>] [-getfacl [-R] <path>
[-getfattr [-R] [-n name | -d] [-e en] <path> [-getmerge [-nl] <src> <localdst>]
[-help [cmd...] ]
[-ls [-d] [-h] [-R] [<path>...] ]
[-mkdir [-p] <path>...] [moveFromLocal <localdst>...<dst>]
[-moveToLocal <src> <localsrc>...<dst>] [-mv <src>...<dst>]
[-put [-f] [-p] [-l] <localsrc>...<dst>]
[-renameSnapshot <snapshotDir> <oldName> <newName>]
[-rm [-f] [-r | -R] [-skipTrash] <src>...]
[-rmdir [--ignore-fail-on-non-empty] <dir>...]
[-setfacl [-R] [{-b | -k} {-m | -x <acl_spec.} <path>] | [--set
<acl_spec> <path>] ]
```

Learn some 12 or 10 commands Shashi

[-setfattr {-n name [-v value] | -x name} <path>] [-setrep [-R] [-w] <rep> <path>...]
[-stat [format] <path>..]
[-tail -[f] <file>]
[-test -[defsz] <path>]
[-text [-ignoreCrc] <src>...] [-touchz <path>...]
[-truncate [-w] <length> <path>...] [-usage [cmd...]]

Generic options supported are

-conf <configuration file> specify an application configuration file
-D <property=value> use value for given property
-fs <local | namenode:port> specify a namenode
-jt <local |resourcemanager:port> specify a ResourceManager
-files <comma separated list of files> specify comma separated files to be copied to the map reduce cluster
-libjars <comma separated list of jars> specify comma separated jar files to include in the classpath.
-archives <comma separated list of archives> specify comma separated archives to be unarchived on the compute machines.

The general command line syntax is

bin/hadoop command [genericoptions] [commadOptions]

List Files in HDFS

To list the files in the root HDFS directory, enter the following command:

$ hdfs dfs -ls /

Found 10 items

drwxrwxrwx	-yarn	hadoop	0 2015-04-28 16:52 /app-logs
drwxr-xr-x	-hdfs	hdfs	0 2015-04-21 14:28 /apps
drwxr-xr-x	-hdfs	hdfs	0 2015-04-21 10:53 /benchmarks
drwxr-xr-x	-hdfs	hdfs	0 2015-04-21 15:18 /hdp
drwxr-xr-x	-mapred	hdfs	0 2015-04-21 14:26 /mapred
drwxr-xr-x	-hdfs	hdfs	0 2015-04-21 14:26 /mr-history
drwxr-xr-x	-hdfs	hdfs	0 2015-04-21 14:27 /system
drwxrwxrwx	-hdfs	hdfs	0 2015-05-07 13:29 /tmp
drwxr-xr-x	-hdfs	hdfs	0 2015-04-27 16:00 /user
drwx-wx-wx	-hdfs	hdfs	0 2015-05-27 9:01 /var

Learn this

To list files in your home directory, enter the following command:$ hdfs dfs -ls

Found 13 items

drwx------	-hdfs	hdfs	0 2015-05-27 20:00 .Trash
drwx------	-hdfs	hdfs	0 2015-05-26 15:43 .staging
drwxr-xr-x	-hdfs	hdfs	0 2015-05-28 13:03 Distributedshell
drwxr-xr-x	-hdfs	hdfs	0 2015-05-14 09:19 TeraGen-50GB
drwxr-xr-x	-hdfs	hdfs	0 2015-05-14 10:11 TeraSort-50GB
drwxr-xr-x	-hdfs	hdfs	0 2015-05-24 20:06 bin
drwxr-xr-x	-hdfs	hdfs	0 2015-04-29 16:52 examples

Learn this

drwxr-xr-x	-hdfs	hdfs	0 2015-04-27 16:00 flume-channel
drwxr-xr-x	-hdfs	hdfs	0 2015-04-29 14:33 oozie-4.1.0
drwxr-xr-x	-hdfs	hdfs	0 2015-04-30 10:35 oozie-examples
drwxr-xr-x	-hdfs	hdfs	0 2015-04-29 20:35 oozie-oozi
drwxr-xr-x	-hdfs	hdfs	0 2015-05-24 18:11 war-and-peace-input
drwxr-xr-x	-hdfs	hdfs	0 2015-05-25 15:22 war-and-peace-output

The same result can be obtained by issuing the following command:

$ hdfs dfs -ls /user/hdfs

Make a Directory in HDFS

To make a directory HDFS, use the following command. As with the -ls command, when no path is supplied, the user's home directory is used (e.g., /users/hdfs).

$ hdfs dfs -mkdir stuff

Copy Files to HDFS

To copy a file from your current local directory into HDFS, use the following command. If a full path is not supplied, your home directory is assumed. In this case, the file test is placed in the directory stuff that was created previously.

$ hdfs dfs -put test stuff

The file transfer can be confirmed by using the -ls command:

$ hdfs dfs -ls stuff

Found 1 items

-rw-r--r--- 2hdfs hdfs 12857 2015-05-29 13:12 stuff/test

Copy Files from HDFS

Files can be copied back to your local file system using the following command. In this case, the file we copied into HDFS, test, will be copied back to the current local directory with the name test-local.

$ hdfs dfs -get stuff/test test-local

Copy Files Within HDFS

The following will copy a file in HFDS:

$ hdfs dfs -cp stuff/test test.hdfs

Delete a File within HDFS

The following command will delete the HDFS file test.dhfs that was created previously:

$ hdfs dfs -rm test.hdfs

Moved: 'hdfs: / /limulus:8020/user/hdfs/stuff/test' to trash at: hdfs: / / limulus:8020 /user/hdfs/ .Trash/Current

Note that when the fs.trash.interval option is set to a non-zero value in core-site .xml, all deleted files are moved to the user's .Trash directory. This can be avoided by including the -skipTrash option.

$ hdfs dfs -rm -skipTrash stuff/test Deleted stuff/test

Delete a Directory in HDFS

The following command will delete the HDFS directory stuff and all its contents:

$ hdfs dfs -rm -r -skipTrash stuff Deleted stuff

Get an HDFS Status Report

Regular users can get an abbreviated HDFS status report using the following command. Those with HDFS administrator privileges will get a full (and potentially long) report. Also, this command uses dfsadmin instead of sfs to invoke administrative commands. The status

report is similar to the data presented in the HDFS web GUI
$ hdfs dfsadmin -report
Configured Capacity: 150340988 1088 (1.37 TB)
Present Capacity: 140794598 1952 (1.28 TB)
DFS Remaining: 125551056 4864 (1.14 TB)
DFS Used: 15243541 7088 (141.97 GB)
DFS Used%: 10.83%
Under replicated blocks: 54 Blocks with corrupt replicas: 0 Missing blocks:0

---

report: Access denied for user deadline. Superuser privilege is required.

b. **Write a short note on running map reduce example and also explain the existing available examples.** **(08 Marks)**

Ans. **Running MapReduce Examples**

All Haddop releases come with MapReduce example applications. Running the existing MapReduce examples is a simple process-once the example files are located, that is. For example, if you installed Haddop version 2.6.0 from the Apache sources under /opt, the example will be in the following directory:
/opt /hadoop-2.6.0/share/hadoop/mapreduce/
In other versions, the examples may be in /urs/lib/hadoop-mapreduce/ or some other location.
The exact location of the example jar file can be found using the find command:
$ find / -name "haddop-mapreduce-example* .jar" -print
Consider the following software environment :

- OS: Linux
- Platform: RHEL 6.6
- Hortonworks HDP 2.2 with Hadoop Version: 2.6

In this environment, the location of the examples is /usr/hdp/2.2.4.2-2/hadoop- mapreduce.
For the purpose of this example, an environment variable called HADOOP_EXAMPLES can be defined as follows:
$ export HADOOP_EXAMPLES=/usr/hdp/2.2.4.2-2/hadoop-mapreduce
Once you define the examples path, you can run the Hadoop examples using the commands discussed in the following sections.
Listing Available Examples
A list of the available examples can be found by running the following command. In some cases, the version number may be part of the jar file (e.g., in the version 2.6 Apache sources, the file is named hadoop-mapreduce-examples-2.6.0.jar).
$ yarn jar SHADOOP_EXAMPLES/hadoop-mapreduce-example.jar
Note: In previous version of Hadoop, the command hadoop jar...was used to run MapReduce programs. Newer versions provides the yarn command, which offers more capabilities. Both commands will work for these examples.
The possible examples are as follows:
An example program must be given as the first argument:
Valid program names are:
aggregate wordcount: An Aggregate based map/reduce program that counts the words in the input files.
aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of the words in the input files.
bbp: A map/reduce program that uses Bailey-Borwein-Plouffe that compute exact bits of Pi.

dbcount: An example job that count the pageview counts from a database.
distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
grep: A map/reduce program that counts the matches of a regex in the input. **join:** A job that effects a join over sorted, equally partitioned datasets
multifilewc: A job that counts words from several files. **pentomino:** A map/reduce tile laying program to find solutions to pentomino problems.
pi: A map/reduce program that estimates Pi using a quasi-MonteCarlo method.
randomtewriter: A map/reduce program that writes 10GB of random textual data per node.
secondarysort: An example defining a secondary sort to the reduce. **aort:** A map/reduce program that sorts the data written by the random writer.
sudoku: A sudoku solver.
teragen: Generate data for the terasort
terasort: Run the terasort
teravalidate: Checking results of terasort
wordcount: A map/reduce program that counts the words in the input files.
wordmean: A map/reduce program that counts the average length of the words in the input files.
wordian: A map/reduce program that counts the median length of the words in the input files.
word standard deviation: A map/reduce program that counts standard deviation of the length of the words in the input files.

OR

2. a. **Explain with neat diagram Apache Hadoop parallel map reduce data flow (or) Explain basic steps of MapReduce parallel data flow with the example of word count program (diagram).** **(08 Marks)**

Ans. **MapReduce Parallel Data Flow:** From a programmers perspective, the MapReduce algorithm is fairly simple. The programmer must provide a mapping function and a reducing function. Operationally, however, the Apache Hadoop parallel MapReduce data flow can be quite complex. Parallel execution of MapReduce requires other steps in addition to the mapper and reducer processes.
The basic steps are as follows:
1.Input Splits, HDFS distributes and replicates data over multiple servers. The default data chunk or block and written to different machines in the cluster. The data are also replicated on multiple machines (typically three machine). These data slices are physical boundaries determined by HDFS and have nothing to do with the data in the file. Also, while not considered part of the MapReduce process, the time required to load and distribute data throughout HDFS servers can be considered part of the total processing time.
The input splits used by MapReduce are logical boundaries based on the input data. For example, the split size can be based on the number of records in a file (if the data exist as records) or an actual size in bytes. Splits are almost always smaller than the HDFS block size. The number of splits corresponds to the number of mapping processes used in the map stage.
2.Map Step. The mapping process is where the parallel nature of Hadoop comes into play. For large amounts of data, many mappers can be operating at the same time. The user provides the specific mapping process. MapReduce will try to execute the mapper on the machines where the block resides. Because the file is replicated in HDFS, the least busy, MapReduce will try to pick a node that is closest to the node that hosts the data block (a characteristic called rack awareness). The last choice is any node in the cluster that has access to HDFS.

3.Combiner Step. It possible to provide an optimization or pre-reduction as part of the map stage where key-value pairs are combined prior to the next stage. The combiner stage is optional.

4.Shuffle Step. Because the parallel reduction stage can complete, all similar keys must be combined and counted by the same reducer process. Therefore, results of the map stage must be collected by the key-value pairs and shuffled to the same reducer process. If only a single reducer process is used, the shuffle stage is not needed.

5.Reduce Step. The final step is the actual reduction. In this stage, the data reduction is performed as per the programmers design. The reduce step is also optional. The results are written to HDFS. Each reducer will write an output file. For example, a MapReduce job running four reducers will create files called part-0000, part-0001, and part-0003.

Figure 1.1 is an example of a simple Hadoop MapReduce data flow for a word count program. The map process counts the words in the split,and the reduce process calculates the total for each word. As mentioned earlier, the actual computation of the map and reduce stages are up to the programmer. The Mapreduce data flow shown in Figure 1.1 is the same regardless of the specific map and reduce tasks.

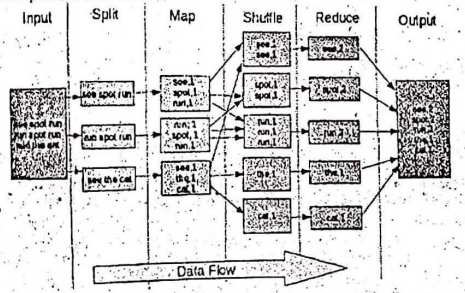


Figure 1.1 Apache Hadoop parallel Mapreduce data flow

The input to the MapReduce application is the following file in HDFS with three lines of text. The goal is to count the number of times each word is used.

see spot run
run spot run
see the cat

The first thing MapReduce will do is create the data splits. For simplicity, each line will be one split. Since each split will require a map task, there are three mapper processes that count the number of words in the split. On a cluster, the results of each map task are written to local disk and not to HDFS. Next, similar keys need to be collected and sent to a reducer process. The shuffle step required data movement and can be expansive in terms of processing time. Depending on the nature of the application, the amount of data that must be shuffle throughout the cluster can vary from small to large.

Once the data have been collected and sorted by key, the reduction step can begin (even if only partial results are available). It is not necessary-and not normally recommended-to have a reducer for each key-value pair as shown in Figure 1.1. In some cases, a single reducer will provide adequate performance; in other cases, multiple reducers may be required to speed up the reduce phase. The number of reducers is a tunable option for many applications. The final

step is to write the output to HDFS.

As mentioned, a combiner step enables some pre-reduction of the map output data. For instance, in the previous example, one map produced the following counts: (run, 1)
(spot, 1)
(run, 1)

As shown in figure 1.2, the count for run can be combined into (run ,2) before the shuffle. This optimization can help minimize of data transfer needed for the shuffle phase.

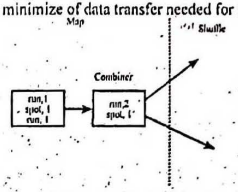


Figure 1.2 Adding a combiner process to the map step in MapReduce.

b. **With two programming ex mapper script and reduces script explain using the Streaming Interface.** **(08 Marks)**

Ans: **Using the Streaming Interface:**

The Apache Hadoop streaming interface enable almost any program to use the MapReduce engine. The streams interface will work with any program that can read and write to stdin and stdout.

When working in the Hadoop streaming mode, only the mapper and the reducer are created by the user. This approach does have the advantage that the mapper and the reducer can be easily tested from the command line. In this example, a Python mapper and reduce, shown in Listings 1.1 and 1.2, will be used.

Listing 1.1 **Python Mapper Script (mappr.py)**

```
#!/ust/bin/env python
import sys
# input comes from STDIN (standard input)
for line in sys.stdin:
# remove leading and trailing whitespace line = line.strip ( )
# split the line into words words = line.split ( )
# increase counters
for word in words:
# write the results to STDOUT (standard output); # what we output here will be the input for
the
# Reduce step, i.e. the input for reducer .py #
# tab-delimited; the trivial word count is 1 print '%s/t%s' % (word, 1)
```

Listing 1.2 **Python Reducer Script (reduce .py)**

```
#!/usr/bin/env python
from operator import itemegetter import sys
current_word=None current_count=0 word = None
#input comes from STDIN
for line in sys.stdin:
# remove leading and trailing whitespace line = line.strip ( )
```

```
# parse the input we got from mapper .py word, count = line .split ('  /t', 1)
# convert count (currently a string) to int
try:
count = int(count)
except ValueError:
# count was not a number, so silently # ignore / discard this line
continue
# this IF-switch only works because Hadoop sorts map output # by key (here: word) before
it is passed to the reducer
if current_word = = word: current_count += count else:
if current_word:
# write result to STUDOUT
print '%s/t%s' 5 (current_word, current_count) current_count = count
current_word =word
# do not forget to output the last word if needed! if current_word = = word:
print '%s/t%s' % (current_word, current_count)
```

The operation of the mapper .py script can be observed by running the commands as shown in the following:

```
$ echo "foo foo quux labs foo bar quux" | ./mapper .py
Foo 1
Foo 1
Quux 1
Labs 1
Foo 1
Bar 1
Quux.1
```

Piping the result of the map into the sort command can create a simulated shuffle phase:

```
Bar 1
Foo 1
Foo 1
Foo 1
Labs 1
Quux 1
Quux 1
```

Finally, the full MapReduce process can be simulated by adding the reducer .py script to the following command pipeline:

```
$ elico "foo foo quux labs foo bar quux" | ./mapper.py | sort
→ -k1, 1 |./reducer .py
Bar 1
Foo 3
Labs 1
Quux 2
```

To run this application using a Hadoop installation, create, if needed, a directory and move the war-and-peace.txt input file into HDFS:

```
$ hdfs dfs -mkdir war-and-peace-input
$ hdfs dfs -put war-and-peace.txt war-and-peace-input
```

Make sure the output directory is removed from any previous test runs:

```
$ hdfs dfs -rm -r -skpiTrash war-and-peace-output
```


Locate the hadoop-streaming.jar file in your distribution. The location may vary, and it may contain a version tag. In this example, the Hortonworks HDP 2.2 distribution was used. The following command line will use the mapper .py and reducer .py to do a word count on the input file.

```
$ hadoop jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar
→
-file ./mapper .py
-mapper ./mapper .py
-file ./reducer .py -reduce ./reducer .py
-input war-and-peace-input/war-and-peace .txt
-output war-and-peace-output
```

The output will be the familiar (_SUCCESS and part -00000) in the war-and-peace- output directory. The actual file name may be slightly difference depending on your Hadoop version. Also note that the Python scripts used in this example could be Bash, Perl, Tcl, Awk, compiled C code,or any language that can read and write from stdin and stdout.

Although the streaming interface is rather simple, it does have some disadvantages over using Java directly. In particular, not all applications are string and character binary data. Another disadvantage is that many tuning parameters available through the full Java Hadoop API are not available in streaming.

Module -2

3. a. Explain How to quire data streams using Apache Flume? (04 Marks)

Ans. Apache Flume is an independent agent designed to collect, transport, and store data into HDFS. Often data transport involves a number of Flume agents that may traverse a series of machines and locations. Flume is often used for log files, social media-generated data, email message, and just about any continuous data source.

As shown in Figure 3.1, a Flume agent is composed of three components.

- **Source.** The source component receives data and sends it to a channel. It can send the data to more than one channel. The input data can be from a real-time source (e.g., weblog) or another Flume agent.
- **Channel.** A channel is a data queue that forwards the source data to the sink destination. It can be though of as buffer that manages input (source) and output (sink) flow rates.
- **Sink.** The sink delivers data to destination such as HDFS, a local file, or another Flume agent.

A Flume agent must have all three of these components defined. A Flume agent can have several sources, channels, and sinks. Sources can write to multiple channels, but a sink can take data from only a single channel. Data written to a channel remain in the channel until a sink removes the data. By default, the data in a channel are kept in memory but may be optionally stored on disk to prevent data loss in the event of a network failure.

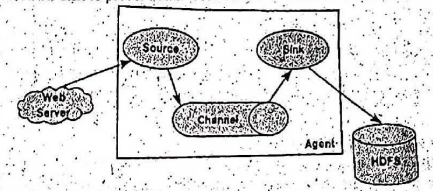


Figure 3.1 Flume Agent with source, channel, and sink(adapted from Apache Flume documentation)

As shown in Figure 3.2, Sqoop agents may be placed in a pipeline, possibly to traverse several machines or domains. This configuration is normally used when data are collected on one machine (e.g., a web server) and sent to another machine that has access to HDFS.

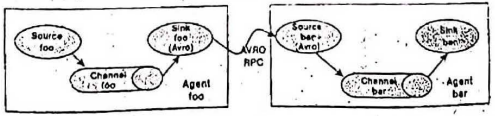


Figure 3.2 Pipeline is created by connecting Fume agents (Adapted from Apache Flume Sqoop Documentation)

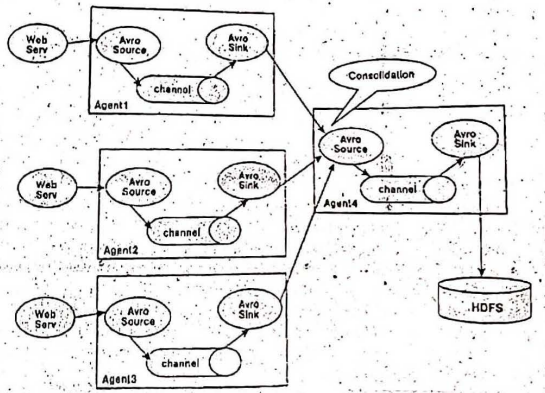


Figure 3.3 A Flume consolidation network (Adapted from Apache Flume Documentation)

In a Flume pipeline, the sink form one agent is connected to the source of another. The data transfer format normally used by Flume, which is called Apache Avro, provides several useful features. First, Avro is a data serialization/deserilization system that uses a compact binary format. The schema is sent as part of the data exchange and is defined using JSON (JavaScript Object Notation). Avro also remote procedure calls (RPCs) to send data. That is, an Avro sink will contact an Avro source to send data. Another useful Flume configuration is shown in Figure 3.3. In this configuration,Flume is used to consolidate several data sources before committing them to HDFS. There are many possible ways to construct Flume transport networks. In addition, other Flume features not described in depth here include plug-ins and interceptors that can enhance Flume pipeline. For pipelines.

b. Explain briefly basic HDFC administration? (12 Marks)

Ans. The NameNode User Interface

Monitoring HDFS can be done in several ways. One of the more convenient ways to get a quick view of HDFS status is through the NameNode user interface. This web-based tool provides essential information about HDFS and offers the capability to browse the HDFS

namespace and logs.

The web-based UI can be started from within Ambari or from a web browser connected to the NameNode. In Ambari, simply select the HDFS service window and click on the Quick Links pull-down menu in the top middle of the page. Select NameNode UI. A new browser tab will open with the UI shown in Figure 3.4. You can also start the UI directly by entering the following command:

$ firefox http:localhost:50070

There are five tabs on the UI : Overview, Datanodes, Snapshot, Startup Progress, and Utilities. The Overview page provides much of the essential information that the command-line tools also offer, but in a much easier- to - read format. The Datanodes tab displays node information like that shown in figure 3.5

The Sanpshot window lists the "snap-shottable" directories and the snapshots. Further information on snapshots can be found in the "HDFS Snapshots" section.

Figure 3.6 provides a NameNode startup progress view. when the NameNode starts, it reads the previous file system image file(fsimage); applies any new edits to the file system image, thereby creating a new file system image; and drops into safe mode until enough DataNodes come online. This progress is shown in real time in the UI as the NameNode starts. Completed phases are displayed in bold text. The currently running phase is displayed in italics. Phases that have not yet begun are displayed in gray text. Figure 3.6, all the phases have been completed, and as indicated in the overview window in Figure 3.4, the system is out of safe mode.

The Utilities menu offers two options. The first, as shown in Figure 3.7, is a file system browser. From this window, you can easily explore the HDFS namespace. The second option, which is not shown, links to the various NameNode logs.

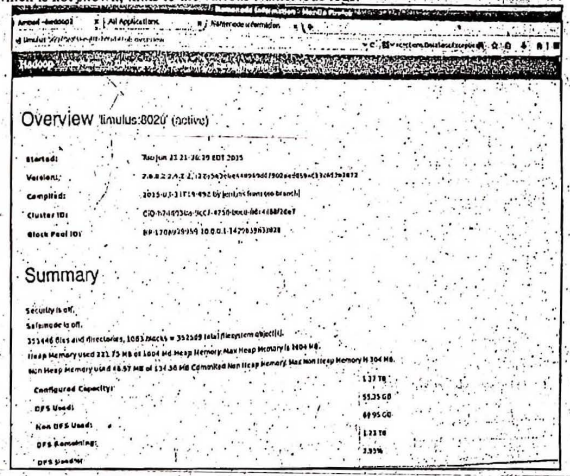


Figure 3.4 Overview page for NameNode user Interface

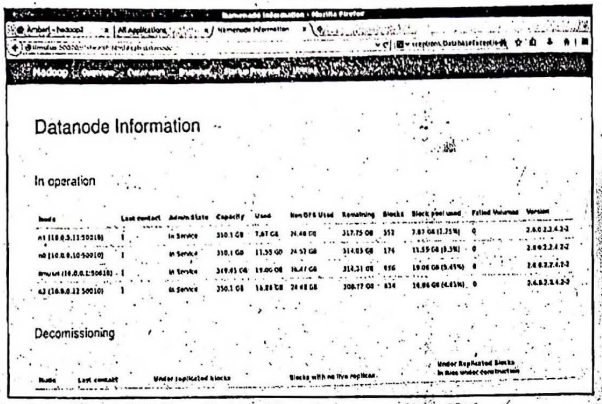


Figure: 3.5 NameNode web interface status of DataNodes

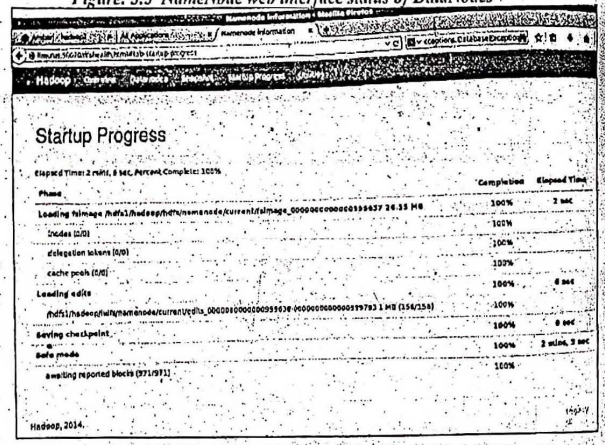


Figure: 3.6 NameNode web interface showing startup progress

Adding Users to HDFS

Keep in mind that errors that crop up while Hadoop applications are running are often due to file permissions.

To quickly create user accounts manually on a Linux-based system, perform thefollowing steps:

1. Add the user to the group for your operating system on the HDFS client system. In most,cases, the groupname should be that of the HDFS superuser, which is often hadoop or hdfs.

useradd G <groupname> <username>

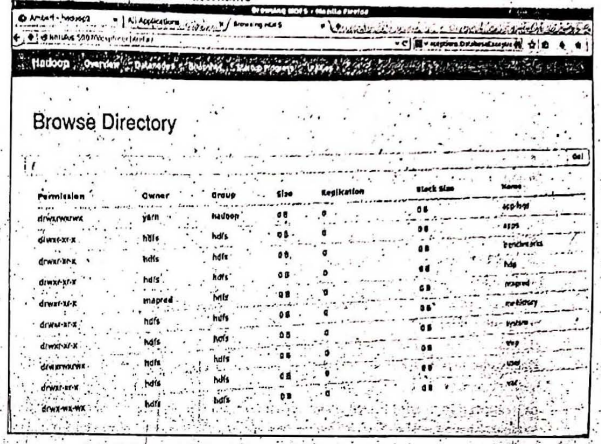


Figure: 3.7 NameNode web interface directory browser

2. Create the username directory in HDFS.

hdfs dfs -mkdir /user/<username>

3. Give that account ownership over its directory in HDFS. hdfs dfs -chown <username>:<groupname> /user/<username> Perform an FSCK on HDFS

To check the health of HDFS, you can issue the hdfs fsck <path> (file system check) command. The entire HDFS namespace can be checked, or a subdirectory can be entered as an argument to the command. The following example checks the entire HDFS namespace.

$ hdfs fsck /

Connecting to namenode via http://limulus:50070

FSCK started by hdfs (auth: SIMPLE) from /10.0.0.1 for path / at Fri May 29 14:48:01 EDT 2015

.....

.....

Status: HEALTHY

Total size: 100433565781 B (Total open files size: 498 B) Total dirs: 20133 1 Total files: 1003

Total symlinks: 0 (Files currently being written: 6)

Total blocks (validated): 1735 (avg. block size 57386781 B) (Total open file blocks (not validated): 6)

Minimally replicated blocks: 173S (100.0 %)

Over-replicated blocks : 0 (0.0 %)

Under - replicated blocs : 0 (0.0 %)

Mis-replicated blocks 0 (0.0 %)

Default replication factor : 2

Avergae block replication : 1.7850144 Corrupt blocks : 0

Missing replicas : 0 (0.0%) Number of data - nodes : 4 Number of racks : 1

FSCK ended at Fri May 29 14 : 48 : 03 EDT 2015 in 1853 milliseconds

The filesystem under path '/' is HEALTHY

Other options provide more detail, include snapshots and open files, and management of corrupted files.

- **move** moves corrupted files to /lost + found
- **delete** deletes corrupted files
- **files** prints out files being checked
- **openforwrite** prints out files opened for writes during clock
- **includeSnapshots** includes snapshot data. The path indicates the existence of a snapshottable directory or the presence of snapshottable directories under it.
- **list-corruptfileblocks** prints out a list of missing blocks and the files to which theybelong.
- **blocks** prints out a block report
- locations prints out locations for every block
- racks prints out network topology for data-node locaitons.

Balancing HDFS

Based on usage patterns and DataNode availability, the number of data blocks across the DataNodes may become unbalanced. To avoid over-utilized DataNodes, the HDFS balancer tool rebalances data blocks across the available DataNodes. Data blocks are moved from over-utilized to under-utilized nodes to within a certain percent threshold. Rebalancing can be done when new DataNodes are added or when a DataNode is removed from service. This step does not create more space in HDFS, but rather improves efficiency.

The HDFS superuser must run the balancer. The simplest way to run the balancei is to enter the following command:

$ hdfs balancer

By default, the balancer will continue to rebalance the nodes until the number of data block on all DataNodes are within 10% of each other. The balancer can be stopped without harming HDFS, at any tune by entering a Ctrl-C. Lower or higher thresholds can be set using the -threshold argument. For example, gming the following command sets a 5% threshold:

$ hdfs balancer -threshold 5

The lower the threshold, the longer the balancer will run. To ensure the balancer does not swamp the cluster networks, you can set a bandwidth limit before running the balancer, as follows:

$ dfsadmin -setBalancerBandwidth newbandwidth

The newbandwidth option is the maximum amount of network bandwidth, in bytes per second, that each DataNode can use during the balancing operation.

Balancing datablocks can also break HBase locality.When HBase regions are moved, some data locality is lost, and the RegionServers will then reqnest the data over the network from remote DataNode(s). This condition will persist until a major HBase compaction event take, place (which ntay either occur at regular intervals or be initiated by the administrator).

HDFS Safe Mode

when the NameNode starts, it loads the file system state from the fsimage and then applies the edits log file. It then waits for DataNodes to report their blocks. During this time, the NameNode stays in a read-only Safe Mode. The NameNode leaves Safe Mode automatically

after the DataNodes have reported that most file system blocks are available.

The administrator can place HDFS in Safe Mode by giving the following command:

$ hdfs dfsadmin -safemode enter

Entering the following command turns off Safe Mode:

$ hdfs dfsadmin -safemode leave

HDFS may drop into Safe Mode if a major issue arises within the file system (e.g., a full DataNode). The file system will not leave Safe Mode until the situation is resolved. To check whether HDFS is in Safe Mode, enter the following command:

$ hdfs dfsadmin -safemode get

Decommissioning HDFS Nodes

If you need to remove a DataNode host/node from the cluster you should decom- mission it first. Assuming the node is responding, it can be easily decommissioned from the Ambari web UI. Simply go to the Hosts view, click on the host and selected Decommission from the pull-down menu next to the DataNode component.

Note that the host may also be acting as a Yarn NodeManager. Use the Ambari H to decommission the YARN host in a similar fashion.

The restoration process is basically a simple copy from the snapshot to the previous directory (or anywhere else). Note the use of the ~/. snapshot/wapi-snap-1 path to restore the file:

$ hdfs dfs -cp /user/hdfs/war-and-peace-input/.snapshot/wapi-snap-1/war-and- peace . txt / user/hdfs/war-and-peace-input

Confirmation that the file has been restored can be obtained by issuing the following command:

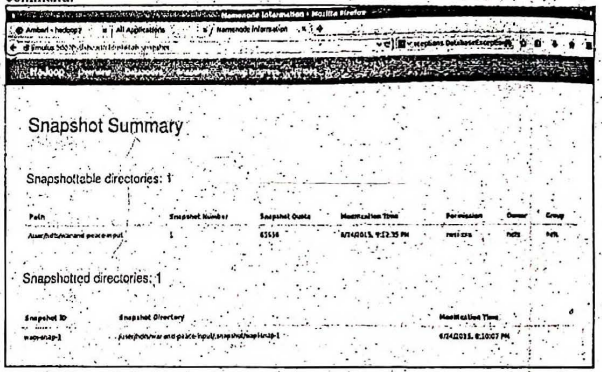


Figure: 3.8 Apache NameNode web interface showing snapshot information

$ hdfs dfs -ls /user/hdfs/war-and-peace-input/ Found 1 items

-rw-r--r-- 2 hdfs hdfs 3288746 2015-06-24 21:12 /user/hdfs/war-and- peace-input/war-and-peace.txt

The NameNode UI provides a listing of snapshottable directories and the snapshots that have been taken. Figure 3.8 shows the results of creating the previous snapshot. To delete a snapshot, give the following command:

$ hdfs dfs -deleteSnapshot /user/hdfs/war-and-peace-input wapi-snap-1

To make a directory "un-snapshottable" (or go back to the default state), use the following command:

$ hdfs dfsadmin -disallowSnapshot /user/hdfs/war-and-peace-input Disallowing snapshot on /user/hdfs/war-and-peace-input succeeded

OR

4. a. How to manage Hadoop service? (08 Marks)

Ans. During the course of normal Hadoop cluster operations, service may fail for any number of reason. Ambari monitors all of the Hadoop service and reports any service interruption to the dashboard. In addition, when the system was installed, an administrative email for the Nagios monitoring system was required. All service interruption notifications are sent to this email address.

Figure 4.1 shows the Ambari dashboard reporting a down DataNode. The service error indicator numbers next to the HDFS service and Hosts menu item indicate this conditions. The DataNode widget also has turned red and indicates that 3/4 DataNode are operating.

Clicking the HDFS service link in the left vertical menu will bring up the service summary screen shoOwn in figure 4.2. The Alters and Health Checks window confirms that a DataNode is down.

The specific host (or hosts) with an issue can be found by examining the Hosts window. As shown in Figure 4.3, the status of host n1 has changed from a green dot with a check mark inside to a yellow dot with a dash inside. An orange dot with a question mark inside indicates the host is not responding and is probably down. Other service interruption indicator may also be set as a result of the unresponsive node.

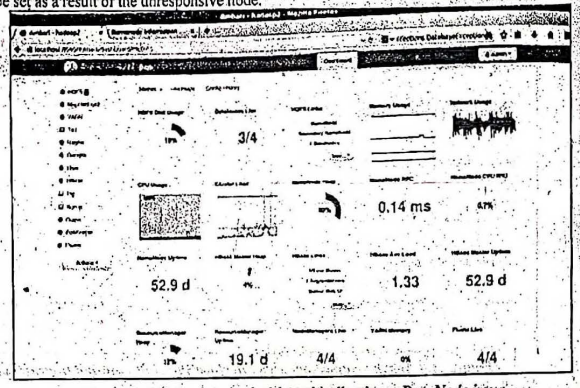


Figure 4.1 Ambari main dashboard indicating a DataNode issue

Clicking on the n1 host link opens the view in Figure 4.4. Inspecting the Components sub-window reveals that the DataNode daemon has stopped on the host. At this point, checking the DataNode logs on host n1 will help identify the actual cause of the failure. Assuming the failures is resolved, the DataNode daemon can be started using the Start option in the pull-down menu next to the service name.

When the DataNode daemon is restarted, a confirmation similar to Figure 4.5 is required from the user.

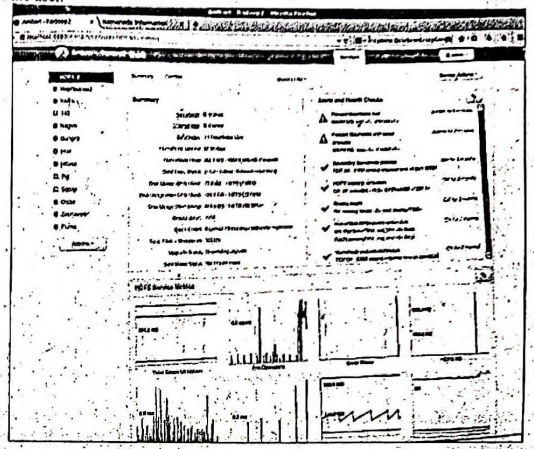


Figure 4.2 Ambari HDFS service summary window indicating a down DataNode

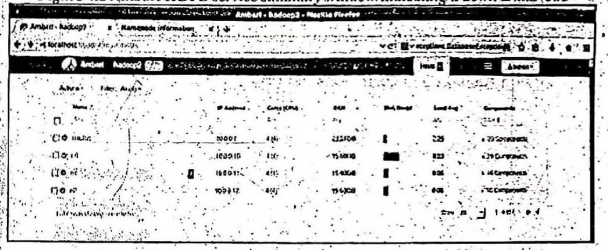


Figure 4.3 Ambari Hosts screen indicating an issue with host n1

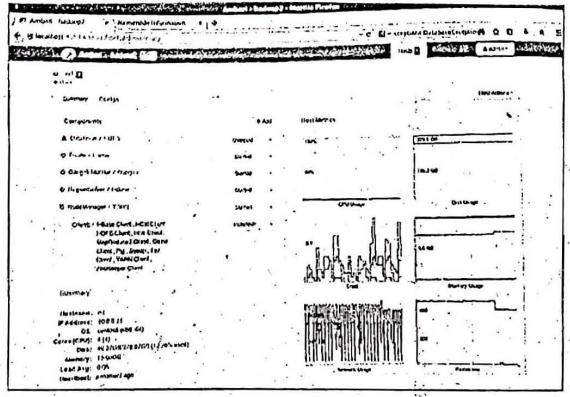


Figure 4.4 Ambari window for host n1 indicating the DataNode/HDFS service has stopped

When a service daemon is started or stopped, a progress window similar to Figure 4.5 is opened. The progress bar indicates the status of each action. Note that previous actions are part of this window. If something goes wrong during the action, the progress bar will turn red. If the system generates a warning about the action, the process bar will turn orange.

When these background operations are running, the small ops (operations) bubble on the top menu bar will indicate how many operations are running.

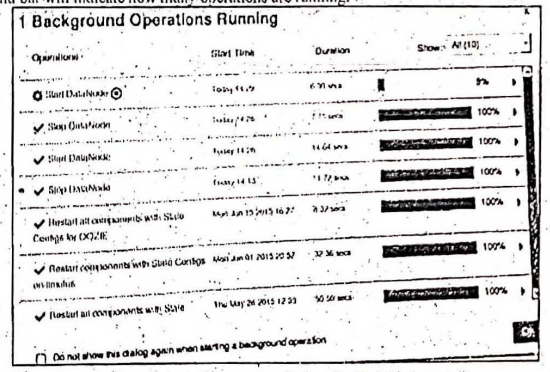


Figure 4.5 Ambari progress window for DataNode restart

Once the DataNode has been restarted successfully, the dashboard will reflect the new status (e.g., 4/4 DataNode are Live). As shown in Figure 4.6, all four

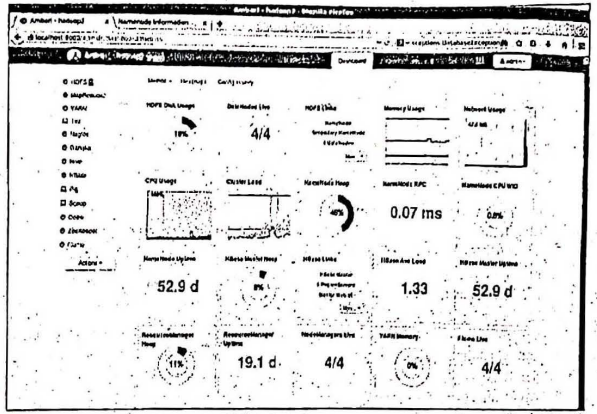


Figure 4.6 Ambari dashboard indicating all DataNodes are running (The service error indicators will slowly drop off the screen) DataNodes are now working and the service error indicators are beginning to slowly disappear. The service error indicators may lag behind the real-time widget updates for several minutes.

b. Write a short note on YARN Application? (04 Marks)

Ans. The central YARN ResourceManager runs as a scheduling daemon on a dedicated machine and acts as the central authority for allocating resources to the various competing applications in the cluster. The ResourceManager has a central and global view of all cluster resources and, therefore, can ensure fairness, capacity, and locality are shared across all users. Depending on the application demand, scheduling priorities, and resource availability, the ResourceManager dynamically allocates resource containers to applications to run on particular nodes. A container is a logical bundle of resources (e.g., memory, cores) bound to a particular cluster node. To enforce and track such assignments, the ResourceManager interacts with a special system daemon running on each node called agers are heartbeat based for scalability. NodeManagers are responsible for local monitoring of resource availability, fault reporting, and container life-cycle management (e.g., starting and killing jobs). The ResourceManager depends on the NodeManagers for its "global view" of the cluster.

User applications are submitted to the ResourceManager via a public protocol and go through an admission control phase during which security credentials are validated and various operational and administrative checks are performed. Those applications that are accepted pass to the scheduler and are allowed to run. Once the scheduler has enough resources to satisfy the request, the application is moved from an accepted state to a running state. Aside from internal bookkeeping, this process ir.volves allocating a container for the single ApplicationMaster and spawning it on a node in the cluster. Often called container 0, the ApplicationMaster does not have any additional resources at this point, but rather must request additional resources from the ResourceManager.

The ApplicationMaster is the "master" user job that manages all application life-cycle aspects, including dynamically increasing and decreasing resource consumption (i.e., containers), managing the flow of execution (e.r., in case of MapReduce jobs, running reducers against the output of maps), handling faults and computation skew, and performing other optimizations. The ApplicationMaster is designed to run arbitrarily user code that can be written in any programming language, as all communication with the ResourceManager and NodeManager is encoded using extensible network protocols.

YARN makes few assumptions about the ApplicationMaster, although in practice it expects most jobs will use a higher-level programming framework. By delegating all these functions to ApplicationMasters, YARN's architecture gains a great deal of scalability, programming model flexibility, and improved user agility. For example, upgrading and testing a new MapReduce framework can be done independently of other running MapReduce frameworks.

Typically, an ApplicationMaster will need to harness the processing power of multiple servers to complete a job. To achieve this, the ApplicationMaster issues resource requests to the ResourceManager. The form of these requests includes specification of locality preferences (e.g., to accommodate HDFS use) and properties of the containers. The ResorceManager will attempt to satisfy the resource requests coming from each application according to availability and scheduling policies. When a resource is scheduled on behalf of an ApplicationMaster, the ResourceManager generates a lease for the resource, which is acquired by a subsequent ApplicationMaster heartbeat.

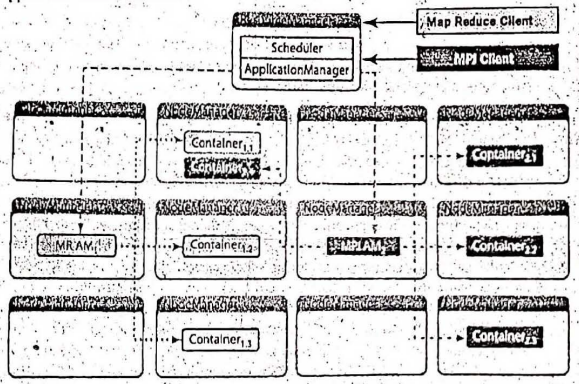


Figure 4.7 : Yarn architecture with two clients(MapReduce and MPI).

The ApplicationMaster then works with the NodeManagers to start the resource. A token-based security mechanism guarantees its authenticity when the ApplicationMaster presents the container lease to the NodeManager. In a typical situation, running containers will communicate with the ApplicationMaster through an application-specific protocol to report status and health information and to receive framework-specific commands. In this way, YARN provides basic infrastructure for monitoring and life-cycle management of containers, while each framework manages application-specific semantics design, in which scheduling

was designed and integrated around managing only MapReduce tasks.

Figure 4.7 illustrates the relationship between the application and YARN components. The YARN components appear as the large outer boxes (ResourceManager and NodeManagers), and the two applications appear as smaller boxes (containers), one dark one light. Each application uses a different ApplicationMaster; the darker client is running a Message passing Interface (MPI) application and the lighter client is running a traditional MapReduce application.

The darker client(MPI AM_2) is running an MPI application,and the lighter client(MR AM_1)is running a MapReduce application.

c. Explain capacity scheduler background. (04 Marks)

Ans. Capacity Scheduler Background

The Capacity scheduler is the default scheduler for YARN that enables multiple groups to securely share a large Hadoop cluster. Developed by the original Hadoop team at Yahoo!, the Capacity scheduler has successfully run many of the largest Hadoop clusters.

To use the Capacity scheduler, one or more queues are configured with a predetermined fraction of the total slot (or processor) capacity. This assignment guarantees a minimum amount of resources for each queue. Administrators can configure soft limits and optional hard limits on the capacity allocated to each queue. Each queue has strict ACLs (Access Control Lists) that control which users can submit applications to individual queues. Also, safeguards are in place to ensure that users cannot view or modify applications from other users.

The Capacity scheduler permits sharing a cluster while giving each user or group certain minimum capacity guarantees. These minimum amounts are not given away in the absence of demand (i.e., a group is always guaranteed a minimum number of resources is available). Excess slots are given to the most starved queues, based on the number of running tasks divided by the queue capacity. Thus, the fullest queues as defined by their initial minimum capacity guarantee get the most needed resources Idle capacity can be assigned and provides elasticity for the users in a cost-effective manner

Administrators can change queue definitions and properties, such as capacity and ACLs, at run time without disrupting users. They can also add more queues at run time, but cannot delete queues at run time. In addition, administrators can stop queues at run time to ensure that while existing applications run to completion, no new appli-cations can be submitted.

The Capacity scheduler currently supports memory-intensive applications, where an application can optionally specify higher memory resource requirements than the default. Using information from the NodeManagers, the capacity scheduler can then place containers on the best-suited nodes.

The capacity scheduler works best when the workloads are well known, which helps in assigning the minimum capacity. For this scheduler to work most effectively, each queue should be assigned a minimal capacity that is less than the maximal expected workload. Within each queue, multiple jobs are scheduled using hierarchical FIFO queues similar to the approach used with the stand-along FIFO scheduler. If there are no queues configured, all jobs are placed in the default queue.

The ResourceManager UI provides a graphical representation of the schedular queues and their utilization. Figure 4.8 shows two jobs running on a four - node cluster. To select the scheduler view, click the scheduler option at the bottom of the left-side vertical menu.

Information on configuring the capacity scheduler can be found at https : //handoop. apache. org/docs/current/hadoop-yarn/hadoop -yarn-site/CapacityScheduler.html and from Apache

Hadoop YARN : Moving beyond MapReduce and Batch Processing with Apache Hadoop 2. In addition to the capacity schedular, Hadoop YARN offers a Fair scheduler. More information can be found on the Hadoop website.

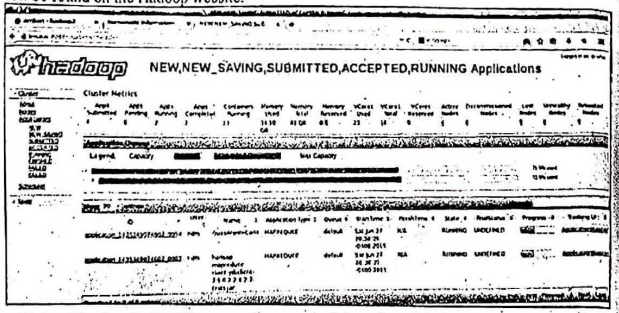


Figure 4.7 : Apache YARN resource manage web interface showing capacity schedules information

Module -3

5. a Describe list of business Intelligence tools used in the organization. Explain any 2 of them used in your organization. (10 Marks)

Ans. According to the list of best business intelligence tools prepared by experts from Finances Online the leading solutions in this category comprise of systems designed to capture, categorize, and analyze corporate data and extract best practices for improved decision making. The more advanced the system is, the more data sources it will combine, including internal metrics coming from different company departments, and external data extracted from third-party systems, social media channels, emails, or even macroeconomic data. Ultimately, business intelligence software helps companies gain insight on their overall growth, sales trends, and customer behavior.

1. Sisense	20. Palo OLAP Server
2. Actuate Business Intelligence and Reporting Tools (BIRT)	21. Pentaho
3. icCube	22. Profit base
4. Domo	23. QlikView
5. Board Management Intelligence Toolkit	24. Rapid insight.
6. Clear Analytics	25. SAP business intelligence
7. Ducen	26. SAP BusinessObjects
8. Gooddata	27. SAP NetWeaver BW
9. IBM Cognos Intelligence	28. SAS BI
10. Insightsquared	29. Silvon
11. JasperSoft	30. Solver
12. Looker	31. SpagoBI

13. Microsoft BI platform	32. SQL Server Analysis Services
14. MicroStrategy	33. Style Intelligence
15. MITS	34. Syntell solutions
16. Openl	35. Targit
17. Oracle BI	36. Vismatica
18. Oracle Enterprise BI Server	37. WebFOCUS
19. Oracle Hyperion System	38. Yellowfin BI

The BI tool used in our organization : Education

As higher education becomes more expensive and competitive, it is a great user of data-based decision-making. There is a strong need for efficiency, increasing revenue, and improving the quality of student experience at all levels of education.

1. Student enrolment (recruitment and retention): Marketing to new potential students requires schools to develop profiles of the students that are most likely to attend. Schools can develop models of what kinds of students are attracted to the school, and then reach out to those students. The students at risk of not returning can be flagged, and corrective measures can be taken in time.

2. Course offerings: Schools can use the class enrolment data to develop models of which new courses are likely to be more popular with students. This can help increase class size, reduce costs, and improve student satisfaction.

3. Alumni pledges: Schools can develop predictive models of which alumni are most likely to pledge financial support to the school. Schools can create a profile for alumni more likely to pledge donations to the school. This could lead to a reduction in the cost of mailings and other forms of outreach to alumni.

b. What are the key elements of a data warehouse? Describe each. (6 Marks)

Ans. DW has four key elements (Figure 5.1).

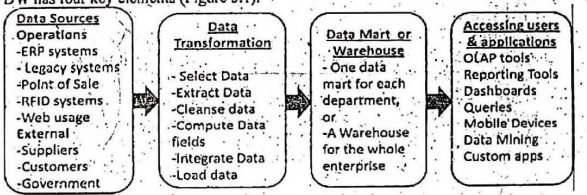


Figure 5.1 Data warehousing architecture

The first element is the data sources that provide the raw data. The second element is the process of transforming that data to meet the decision needs. The third element is the methods of regularly and accurately loading of that data into EDW or data marts. The fourth element is the data access and analysis part, where devices and applications use the data from DW to deliver insights and other benefits to users.

Data Sources

DWs are created from structured data sources. Unstructured data, such as text data, would need to be structured before inserted into DW.

1. Operations data include data from all business applications, including from ERPs systems that form the backbone of an organization's IT systems. The data to be extracted will depend upon the subject matter of DW. For example, for a sales/marketing DW, only the data about customers, orders, customer service, and so on would be extracted.
2. Other applications, such as point-of-sale (POS) terminals and e-commerce applications, provide customer-facing data. Supplier data could come from supply chain management systems. Planning and budget data should also be added as needed for making comparisons against targets.
3. External syndicated data, such as weather or economic activity data, could also be added to DW, as needed, to provide good contextual information to decision makers.

Figure 5.2 Data warehousing architecture

Data Transformation Processes.

The heart of a useful DW is the processes to populate the DW with good quality data. This is called the extract-transform-load (ETL) cycle.

1. Data should be extracted from many operational (transactional) database sources on a regular basis.
2. Extracted data should be aligned together by key fields. It should be cleansed of any irregularities or missing values. It should be rolled up together to the same level of granularity. Desired fields, such as daily sales totals, should be computed. The entire data should then be brought to the same format as the central table of DW.
3. The transformed data should then be uploaded into DW. This ETL process should be run at a regular frequency. Daily transaction data can be extracted from ERPs, transformed, and uploaded to the database the same night. Thus, DW is up-to-date next morning. If DW is needed for near-real-time information access, then the ETL processes would need to be executed more frequently. ETL work is usually automated using programing scripts that are written, tested, and then deployed for periodic updating DW.

DW Design

Star schema is the preferred data architecture for most DWs. There is a central fact table that provides most of the information of interest. There are lookup tables that provide detailed values for codes used in the central table. For example, the central table may use digits to represent a sales person. The lookup table will help provide the name for that sales person code. Here is an example of a star schema for a data mart for monitoring sales performance (Figure5.2).

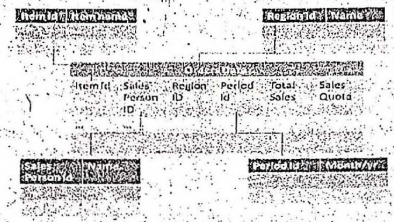


Figure 5.2 Star schema architecture

Other schemas include the snowflake architecture. The difference between a star and snowflake is that in the latter, the lookup tables can have their own further lookup tables.

There are many technology choices for developing DW. This includes selecting the right

database management system and the right set of data management tools. There are a few big and reliable providers of DW systems.

The provider of the operational DBMS may be chosen for DW also.

Alternatively, a best-of-breed DW vendor could be used. There are also a variety of tools out there for data migration, data upload, data retrieval, and data analysis.

DW Access

Data from DW could be accessed for many purposes, through many devices.

1. A primary use of DW is to produce routine management and monitoring reports. For example, a sales performance report would show sales by many dimensions, and compared with plan. A dashboarding system will use data from the warehouse and present analysis to users. The data from DW can be used to populate customized performance dashboards for executives. The dashboard could include drill-down capabilities to analyze the performance data for root cause analysis.
2. The data from the warehouse could be used for ad hoc queries and any other applications that make use of the internal data.
3. Data from DW is used to provide data for mining purposes. Parts of the data would be extracted, and then combined with other relevant data, for data mining

OR

6. a. **Describe the key steps in the data mining process. Why is it important to follow these processes?** **(08 Marks)**

Ans: Effective and successful use of data mining activity requires both business and technology skills. The business aspects help understand the domain and the key questions. It also helps one imagine possible relationships in the data and create hypotheses to test it. The IT aspects help fetch the data from many sources, clean up the data, assemble it to meet the needs of the business problem, and then run the data mining techniques on the platform.

An important element is to go after the problem iteratively. It is better to divide and conquer the problem with smaller amounts of data, and get closer to the heart of the solution in an iterative sequence of steps. There are several best practices learned from the use of data mining techniques over a long period of time. The data mining industry has proposed a Cross-Industry Standard Process for Data Mining (CRISP-DM). It has six essential steps (Figure 6.1):

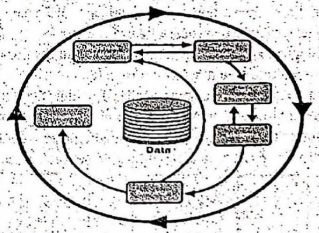


Figure 6.1 CRISP-DM data mining cycle

1. The first and most important step in data mining is business understanding that is, asking the right business questions. A question is a good one if answering it would lead to large payoffs for the organization, financially and otherwise. In other words, selecting a data mining project is like any other project, in which it should show strong payoffs if the project

is successful. There should be strong executive support for the data mining project, which means that the project aligns well with the business strategy.

2. A second important step is to be creative and open in proposing imaginative hypotheses for the solution. Thinking outside the box is important, both in terms of a proposed model as well in the data sets available and required.

3. The data should be clean and of high quality. It is important to assemble a team that has a mix of technical and business skills, who understand the domain and the data. Data cleaning can take 60 to 70 percent of the time in a data mining project. It may be desirable to add new data elements from external sources of data that could help improve predictive accuracy.

4. Patience is required in continuously engaging with the data until the data yields some good insights. A host of modeling tools and algorithms should be used. A tool could be tried with different options, such as running different decision tree algorithms.

5. One should not accept what the data says at first. It is better to triangulate the analysis by applying multiple data mining techniques and conducting many what-if scenarios, to build confidence in the solution. Evaluate the model's predictive accuracy with more test data.

6. The dissemination and rollout of the solution is the key to project success. Otherwise the project will be a waste of time and will be a setback for establishing and supporting a data-based decision-process culture in the organization. The model should be embedded in the organization's business processes.

b. What are the data visualization techniques? When would you use tables or graphs? (08 Marks)

Ans. Data Visualization techniques are :

1.Pixel oriented visualization techniques:

A simple way to visualize the value of a dimension is to use a pixel where the color of the pixel reflects the dimension's value.

For a data set of m dimensions pixel oriented techniques create m windows on the screen, one for each dimension.

The m dimension values of a record are mapped to m pixels at the corresponding position in the windows. The color of the pixel reflects other corresponding values.

Inside a window, the data values are arranged in some global order shared by all windows

Eg: All Electronics maintains a customer information table, which consists of 4 dimensions: income, credit_limit, transaction_volume and age. We analyze the correlation between income and other attributes by visualization.

We sort all customers in income in ascending order and use this order to layout the customer data in the 4 visualization windows as shown in fig.

The pixel colors are chosen so that the smaller the value, the lighter the shading.

Using pixel based visualization we can easily observe that credit_limit increases as income increases customer whose income is in the middle range are more likely to purchase more from All Electronics, these is no clear correlation between income and age.

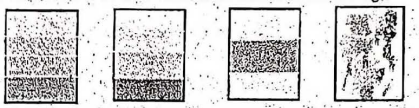


Fig 6.2 : Pixel oriented visualization of 4 attributes by sorting all customers in income Ascending order.

2.Geometric Projection visualization techniques

A drawback of pixel-oriented visualization techniques is that they cannot help us much in understanding the distribution of data in a multidimensional space.

Geometric projection techniques help users find interesting projections of multidimensional data sets.

A scatter plot displays 2-D data point using Cartesian co-ordinates. A third dimension can be added using different colors of shapes to represent different data points.

Eg. Where x and y are two spatial attributes and the third dimension is represented by different shapes

Through this visualization, we can see that points of types "+" &"X" tend to be collocated.

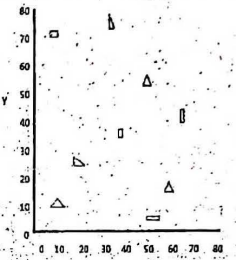


Fig 6.3 : visualization of 2D data set using scatter plot

3.Icon based visualization techniques:-

It uses small icons to represent multidimensional data values

2 popular icon based techniques:-

3.1 Chern off faces: - They display multidimensional data of up to 18 variables as a cartoon human face.

Fig 6.4 : chern off faces each face represents an 'n' dimensional data points (n<18)

$\hspace{1 cm}$ 3.2 Stick figures: It maps multidimensional data to five –piece stick figure, where each figure has 4 limbs and a body.

2 dimensions are mapped to the display axes and the remaining dimensions are mapped to the angle and/ or length of the limbs.

4.Hierarchical visualization techniques (i.e. subspaces)

The subspaces are visualized in a hierarchical manner.

Types of Charts

There are many kinds of data as seen in the caselet above. Time series data is the most popular form of data. It helps reveal patterns over time. However, data could be organized around alphabetical list of things, such as countries or products or salespeople. Figure 6.1 shows some of the popular chart types and their usage.

1. **Line graph.** This is a basic and most popular type of displaying information. It shows data as a series of points connected by straight line segments. If mining with time-series data, time is usually shown on the x-axis. Multiple variables can be represented on the same scale on

y-axis to compare of the line graphs of all the variables.

2. **Scatter plot:** This is another very basic and useful graphic form. It helps reveal the relationship between two variables. In the above caselet, it shows two dimensions: Life Expectancy and Fertility Rate. Unlike in a line graph, there are no line segments connecting the points.

3. **Bar graph:** A bar graph shows thin colorful rectangular bars with their lengths being proportional to the values represented. The bars can be plotted vertically or horizontally. The bar graphs use a lot of more ink than the line graph and should be used when line graphs are inadequate.

4. **Stacked Bar graphs:** These are a particular method of doing bar graphs. Values of multiple variables are stacked one on top of the other to tell an interesting story. Bars can also be normalized such as the total height of every bar is equal, so it can show the relative composition of each bar.

5. **Histograms:** These are like bar graphs, except that they are useful in showing data frequencies or data values on classes (or ranges) of a numerical variable.

6. **Pie charts:** These are very popular to show the distribution of a variable, such as sales by region. The size of a slice is representative of the relative strengths of each value.

7. **Box charts:** These are special form of charts to show the distribution of variables. The box shows the middle half of the values, while whiskers on both sides extend to the extreme values in either direction.

8. **Bubble Graph:** This is an interesting way of displaying multiple dimensions in one chart. It is a variant of a scatter plot with many data points marked on two dimensions. Now imagine that each data point on the graph is a bubble (or a circle) ... the size of the circle and the color fill in the circle could represent two additional dimensions.

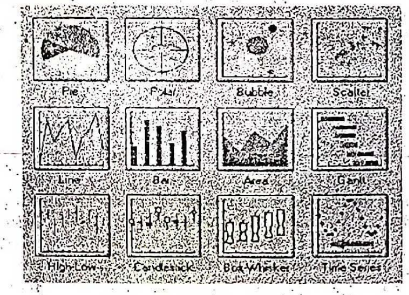


Figure 6.5: Many types of graphs

9. **Dials:** These are charts like the speed dial in the car, that shows whether the variable value (such as sales number) is in the low range, medium range, or high range. These ranges could be colored red, yellow and green to give an instant view of the data.

10. **Geographical Data** maps are particularly useful maps to denote statistics. Figure 6.6 shows a tweet density map of the US. It shows where the tweets emerge from in the US.

11. **Pictographs:** One can use pictures to represent data. E.g. Figure 6.7 shows the number of liters of water needed to produce one pound of each of the products, where images are used to show the product for easy reference. Each droplet of water also represents 50 liters of water.

Figure 6.6 : US tweet map (Source: Slate.com)

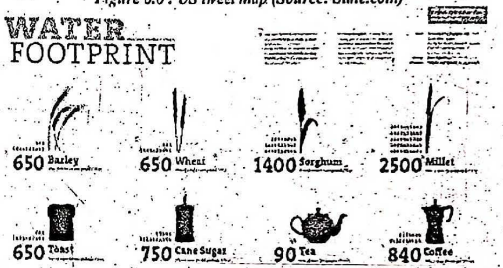


Figure 6.7: Pictograph of Water footprint (source : waterfootprint.org)

A table is best when:
- You need to look up specific values
- Users need precise values
- You need to precisely compare related values
- You have multiple data sets with different units of measure

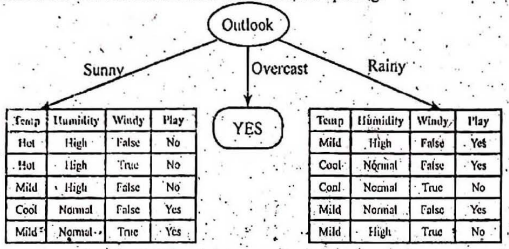
A graph is best when:
- The message is contained in the shape of the values
- You want to reveal relationships among multiple values (similarities and differences)
- Show general trends
- You have large data sets
- Graphs and tables serve different purposes. Choose the appropriate data display to fit your purpose.

Module -4

7. a. **What is a splitting variable? Describe three criteria for choosing splitting variable.** **(08 Marks)**

Ans. Splitting the Tree: From the root node, the decision tree will be split into three branches or sub-trees, one for each of the three values of outlook. Data for the root node (the entire data) will be divided into the three segments, one for each of the value of outlook. The sunny branch will inherit the data for the instances that had sunny as the value of outlook. These will be used for further building of that sub-tree. Similarly, the rainy branch will inherit data for the instances that had rainy as the value of outlook. These will be used for further building of that sub-tree. The overcast branch will inherit the data for the instances that had overcast as the outlook. However, there will be no need to build further on that branch. There is a clear decision, yes, for all instances when outlook value is overcast.

The decision tree will look like this after the first level of splitting.



Temp	Humidity	Windy	Play
Hot	High	False	No
Hot	High	True	No
Mild	High	False	No
Cool	Normal	False	Yes
Mild	Normal	True	Yes

Temp	Humidity	Windy	Play
Mild	High	False	Yes
Cool	Normal	False	Yes
Cool	Normal	True	No
Mild	Normal	False	Yes
Mild	High	True	No

Decision trees employ the divide and conquer method. The data is branched at each node according to certain criteria until all the data is assigned to leaf nodes. It recursively divides a training set until each division consists of examples from one class.

The following is a pseudo code for making decision trees:

1. Create a root node and assign all of the training data to it.
2. Select the best splitting attribute according to certain criteria.
3. Add a branch to the root node for each value of the split.
4. Split the data into mutually exclusive subsets along the lines of the specific split.
5. Repeat steps 2 and 3 for each and every leaf node until a stopping criteria is reached.

There are many algorithms for making decision trees. The most popular ones are C5, CART, and CHAID. They differ on three key elements:

1. *Splitting criteria*

a. Which variable to use for the first split? How should one determine the most important variable for the first branch, and subsequently,

for each subtree? There are many measures like least errors, information gain, and Gini coefficient.

b. What values to use for the split? If the variables have continuous values, such as for age or BP, what value-ranges should be used to make bins?

c. How many branches should be allowed for each node? There could be binary trees, with just two branches at each node. Or there could be more branches allowed.

2. *Stopping criteria*

a. When to stop building the tree? There are two major ways to make that determination. The tree building could be stopped when a certain depth of the branches has been reached and the tree becomes unreadable after that. The tree could also be stopped when the error level at any node is within predefined tolerable levels.

3. *Pruning*: The tree could be trimmed to make it more balanced and more easily usable. The pruning is often done after the tree is constructed, to balance out the tree and improve usability. The symptoms of an overfitted tree are a tree too deep, with too many branches, some of which may reflect anomalies due to noise or outliers. Thus, the tree should be pruned. There are two approaches to avoid over-fitting.

b. **Compare and contrast decision trees with regression models?** **(08 Marks)**

Ans. **Advantages and Disadvantages of Regression Models**

Regression models are very popular because they offer many advantages.

1. Regression models are easy to understand as they are built upon basic statistical principles, such as correlation and least square error.
2. Regression models provide simple algebraic equations that are easy to understand and use.
3. The strength (or the goodness of fit) of the regression model is measured in terms of the correlation coefficients, and other related statistical parameters that are well understood.
4. Regression models can match and beat the predictive power of other modeling techniques.
5. Regression models can include all the variables that one wants to include in the model.
6. Regression modeling tools are pervasive. They are found in statistical packages as well as data mining packages. MS Excel spreadsheets can also provide simple regression modeling capabilities.

Regression models can however prove inadequate under many circumstances.

1. Regression models cannot cover for poor data quality issues. If the data is not prepared well to remove missing values, or is not well-behaved in terms of a normal distribution, the validity of the model suffers.
2. Regression models suffer from collinear problems (meaning strong linear correlations among some independent variables). If the independent variables have strong correlations among themselves, then they will eat into each other's predictive power and the regression coefficients will lose their ruggedness.
3. Regression models will not automatically choose between highly collinear variables, although some packages attempt to do that. Regression models can be unwieldy and unreliable if a large number of variables are included in the model. All variables entered into the model will be reflected in the regression equation, irrespective of their contribution to the predictive power of the model. There is no concept of automatic pruning the model.
4. Regression models do not automatically take care of nonlinearity.

The user needs to imagine the kind of additional terms that might be needed to be added to the regression model to improve its fit.

5. Regression models work only with numeric data and not with categorical variables. There are ways to deal with categorical variables though by creating multiple new variables with a yes/no value.

OR

8. a. **Compare Neural network with decision tree. Give the advantage and disadvantages of neural network.** **(08 Marks)**

Ans. There are many differences between these two, but in practical terms, there are three main things to consider: speed, interpretability, and accuracy.

Decision Trees

Should be faster once trained (although both algorithms can train slowly depending on exact algorithm and the amount/dimensionality of the data). This is because a decision tree inherently "throws away" the input features that it doesn't find useful, whereas a neural net will use them all unless you do some feature selection as a pre-processing step.

If it is important to understand what the model is doing, the trees are very interpretable.

Only model functions which are axis-parallel splits of the data, which may not be the case.

You probably want to be sure to prune the tree to avoid over-fitting.

Neural Nets

Slower (both for training and classification), and less interpretable.

If your data arrives in a stream, you can do incremental updates with stochastic gradient descent (unlike decision trees, which use inherently batch-learning algorithms).

Can model more arbitrary functions (nonlinear interactions, etc.) and therefore might be more accurate, provided there is enough training data. But it can be prone to over-fitting as well.

There are many advantages of using ANN.

1. ANNs impose very little restrictions on their use. ANN can deal with (identify/model) highly nonlinear relationships on their own, without much work from the user or analyst. They help find practical data-driven solutions where algorithmic solutions are nonexistent or too complicated.
2. There is no need to program ANN neural networks, as they learn from examples. They get better with use, without much programing effort.
3. ANN can handle a variety of problem types, including classification, clustering, associations, and so on.
4. ANNs are tolerant of data quality issues, and they do not restrict the data to follow strict normality and/or independence assumptions.
5. ANN can handle both numerical and categorical variables.
6. ANNs can be much faster than other techniques.
7. Most importantly, ANN usually provide better results (prediction and/or clustering) compared to statistical counterparts, once they have been trained enough.

The key disadvantages arise from the fact that they are not easy to interpret or explain or compute.

1. They are deemed to be black-box solutions, lacking explainability.
2. Optimal design of ANN is still an art: It requires expertise and extensive experimentation.
3. It can be difficult to handle a large number of variables (especially the rich nominal attributes) with an ANN.
4. It takes large data sets to train an ANN.

b. Define Clusters? Describe three business applications in your industry where cluster analysis will be useful. (08 Marks)

Ans. Definition of a Cluster :An operational definition of a cluster is that, given a representation of n objects, find K groups based on a measure of similarity, such that objects within the same group are alike but the objects in different groups are not alike.

However, the notion of similarity can be interpreted in many ways. Clusters can differ in terms of their shape, size, and density. Clusters are patterns, and there can be many kinds of patterns. Some clusters are the traditional types, such as data points hanging together. However, there are other clusters, such as all points representing the circumference of a circle. There may be concentric circles with points of different circles representing different clusters. The presence of noise in the data makes the detection of the clusters even more

difficult.

An ideal cluster can be defined as a set of points that is compact and isolated.

In reality, a cluster is a subjective entity whose significance and interpretation requires domain knowledge. In the sample data below (Figure 8.1), how many clusters can one visualize?

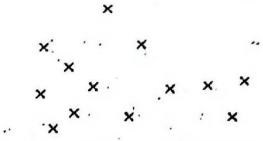


Figure 8.1: Visual cluster example

It seems like there are two clusters of approximately equal sizes. However, they can be seen as three clusters, depending on how we draw the dividing lines. There is not a truly optimal way to calculate it. Heuristics are often used to define the number of clusters.

Three business applications:

Cluster analysis is used in almost every field where there is a large variety of transactions. It helps provide characterization, definition, and labels for populations. It can help identify natural groupings of customers, products, patients, and so on. It can also help identify outliers in a specific domain and thus decrease the size and complexity of problems. A prominent business application of cluster analysis is in market research. Customers are segmented into clusters based on their characteristics—wants and needs, geography, price sensitivity, and so on. Here are some examples of clustering:

1. *Market Segmentation*: Categorizing customers according to their similarities, for instance by their common wants and needs, and propensity to pay, can help with targeted marketing.
2. *Product portfolio*: People of similar sizes can be grouped together to make small, medium and large sizes for clothing items.
3. *Text Mining*: Clustering can help organize a given collection of text documents according to their content similarities into clusters of related topics.

Module -5

9. a. Give the Comparison between Text Mining and Data Mining? (08 Marks)

Ans. Text Mining is a form of data mining. There are many common elements between Text and Data Mining. However, there are some key differences (Table Below). The key difference is that text mining requires conversion of text data into frequency data, before data mining techniques can be applied.

Dimension	Text Mining	Data Mining
Nature of data	Unstructured data: words, phrases, sentences	Numbers; alphabetical and logical values
Language used	Many languages and dialects used in the world; many languages are extinct, new documents are discovered	Similar numerical systems across the world
Clarity and precision	Sentences can be ambiguous; sentiment may contradict the words	Numbers are precise
Consistency	Different parts of the text can contradict each other	Different parts of data can be inconsistent; thus, requiring statistical significance analysis

Sentiment	Text may present a clear and consistent or mixed sentiment, across a continuum. Spoken words adds further sentiment	N/A
Quality	Spelling errors. Differing values of proper nouns, such as names. Varying quality of language translation	Issues with missing values, outliers, and so on
Nature of analysis	Keyword-based search; coexistence of themes; sentiment mining	A full wide range of statistical and machine-learning analysis for relationships and differences

b. In what ways is Naive-Bayes better than other classification techniques? Compare with decision tree. (08 Marks)

Ans. Classification is the separation or ordering of objects into classes .There are two phases in classification algorithm: first, the algorithm tries to find a model for the class attribute as a function of other variables of the datasets. Next, it applies previously designed model on the new and unseen datasets for determining the related class of each record .

Classification has been applied in many fields such as medical, astronomy, commerce, biology, media, etc. There are many techniques in classification method like: Decision Tree, Naïve Bayes, k-Nearest Neighbor, Neural Networks, Support Vector Machine, and Genetic Algorithm. In this paper we will use Decision Tree, Naïve Bayes, and k-Nearest Neighbor. They are both supervised learning algorithms used for classification tasks. It strongly depends of the data you have and what you are trying to learn .

Although it depends on the problem you are solving, but some general advantages are following:

Naive - Bayes :

1. Work well with small dataset compared to DT which need more data
2. Lesser overfitting
3. Smaller in size and faster in processing

Decision Tree:

1. Decision Trees are very flexible, easy to understand, and easy to debug
2. No preprocessing or transformation of features required
3. Prone to overfitting but you can user pruning or Random forests to avoid that.

In Brief : ***Decision Tree***

A decision tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. The popular Decision Tree algorithms are ID3, C4.5, CART. The ID3 algorithm is considered as a very simple decision tree algorithm. It uses information gain as splitting criteria. C4.5 is an evolution of ID3. It uses gain ratio as splitting criteria.

CART algorithm uses Gini coefficient as the test attribute selection criteria, and each time selects an attribute with the smallest Gini coefficient as the test attribute for a given set [5].

The advantage of using Decision Trees in classifying the data is that they are simple to understand and interpret . However, decision trees have such disadvantages as :

1) Most of the algorithms (like ID3 and C4.5) require that the target attribute will have only discrete values.

2) As decision trees use the "divide and conquer" method, they tend to perform well if a few highly relevant attributes exist, but less so if many complex interactions are present.

Naive Bayes : Naive Bayesian classifiers assume that there are no dependencies amongst attributes. This assumption is called class conditional independence. It is made to simplify

the computations involved and, hence is called "naive" . This classifier is also called idiot Bayes, simple Bayes, or independent Bayes .

The advantages of Naive Bayes are :

- It uses a very intuitive technique. Bayes classifiers, unlike neural networks, do not have several free parameters that must be set. This greatly simplifies the design process.
- Since the classifier returns probabilities, it is simpler to apply these results to a wide variety of tasks than if an arbitrary scale was used.
- It does not require large amounts of data before learning can begin.
- Naive Bayes classifiers are computationally fast when making decisions.

OR

10. a. What is clickstream analysis? (02 Marks)

Ans. On a Web site, clickstream analysis (also called clickstream analytics) is the process of collecting, analyzing and reporting aggregate data about which pages a website visitor visits -- and in what order. The path the visitor takes though a website is called the clickstream.

There are two levels of clickstream analysis, traffic analytics and e-commerce analtyics. Traffic analytics operates at the server level and tracks how many pages are served to the user, how long it takes each page to load, how often the user hits the browser's back or stop button and how much data is transmitted before the user moves on. E-commerce-based analysis uses clickstream data to determine the effectiveness of the site as a channel-to-market. It's concerned with what pages the shopper lingers on, what the shopper puts in or takes out of a shopping cart, what items the shopper purchases, whether or not the shopper belongs to a loyalty program and uses a coupon code and the shopper's preferred method of payment.

Because an extremely large volume of data can be gathered through clickstream analysis, many e-businesses rely on big data analytics and related tools such as Hadoop to help interpret the data and generate reports for specific areas of interest. Clickstream analysis is considered to be most effective when used in conjunction with other, more traditional, market evaluation resources.

b. Explain briefly the techniques and algorithms of social network analysis? (14 Marks)

Ans. **TECHNIQUES AND ALGORITHM :** There are two major levels of social network analysis - discovering sub-networks within the network and ranking the nodes to find more important nodes or hubs.

Finding Sub-networks: A large network could be the better analyzed and managed if it can be seen as an interconnected set of distinct sub-networks each with its own distinct identity and unique characteristics. This is like doing a cluster analysis of nodes. Nodes with strong ties between them would belong to the same sub-network ,while those with weak or no ties would belong to separate sub-networks .This is unsupervised learning technique ,as in Apriori there is no correct number of sub-networks in a network. The usefulness of the of the network structure for decision -making is the main criterion for adopting a particular structure.

Visual representation of networks can help identify sub-networks. Use of color can help differentiate the types of nodes. Representing strong ties with thicker or bolder lines could help visually identify the stronger relationships. A sub-network could be a collection of strong relationships around a hub node. In this case , the hub node could represent a distinct sub-network. A sub-network could also be a subset of nodes with dense relationships between them. In this case , one or more nodes will act as gateway to the rest of the network.

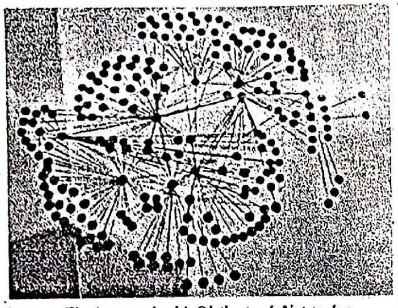


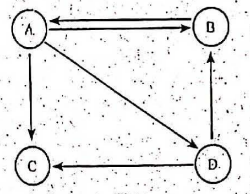
Fig A network with Distinct sub Network

Computing Importance of Nodes: When the connections between nodes in the network have a direction to them ,then the nodes can be compared for their relative influence or rank. This is done using 'Influence Flow Model'. Every outbound link from a node can be considered an outflow of influence. Every incoming link is similar an inflow of influence. More in-links to a node means greater importance. Thus there will be many direct and indirect flows of influence between any two nodes in the network.

Computing the relative influence of each node is done on the basis of an input-output matrix of flows of influence among the nodes. Assume each nodes has an influence value . The computational task is to identity a set of rank values that satisfies the set of links between the nodes. It is an iterative task where we begin with some initial values and continue to iterate till the rank values stabilize.

Consider the following simple network with 4 nodes (A,B,C,D) and 6 directed links between them as shown in the figure(10.2). Note that there is a bidirectional link . Here are the links:

Node A links into B
Node B links into C
Node C links into D
Node D links into A
Node A links into C
Node B links into A



Fig

The goal is to find the relative importance , or rank , or every node in the network . This will help identify the most importance node(s) in the network.

We begin by assigning the variables for influence (or rank) value for each node , as Ra , Rb , Rc , and Rd. The goal is to find the relative values of these variables.

There are two outbound links from node A to node B and C. Thus , both B and C receives half of node A's influence. Similarly, there are two outbound links from node B to node C and A ,so both C and A receives half of node B's influence.

There is only outbound link from node D to node A . Thus ,node A gets all the influence of node D. There is only outbound link from node C to node D and hence, node D gets all the influence of node C.

Node A gets all of the influence of node D and half the influence of node B.

Thus , $Ra = 0.5 \times Rb + Rd.$

Node B gets half the influence of node A.

Thus, $Rc = 0.5 \times Ra.$

Node C gets half the influence of node A and half the influence of node B.

Thus, $Rc = 0.5 \times Ra + 0.5 \times Rb.$

Node D gets all of the influence of node C and half the influence of node B.

Thus, $Rd = Rc.$

We have 4 equations using 4 variables . These can be solved mathematically.

We can represent the coefficient of these 4 equations in a matrix form as shown in the Dataset (10.1) given below..This is the Influence Matrix . The zero value represent that the term is not represented in an equation.

Data Set 10.1

	Ra	Rb	Rc	Rd
Ra	0	0.50	0	1.00
Rb	0.50	0	0	0
Rc	0.50	0.50	0	0
Rd	0	0	1.00	0

For simplification , let us also state that all the rank values add up to 1. Thus , each node has a fraction as the rank value . Let us start with an initial set of rank values and then iteratively compute new rank values till they stabilize. One can start with any initial rank values, such as *1/n* or *1/4* for the nodes.

Variable	Initial Value
Ra	0.250
Rb	0.250
Rc	0.250
Rd	0.250

Variable	Initial Value	Iteration 1
Ra	0.250	0.375
Rb	0.250	0.125
Rc	0.250	0.250
Rd	0.250	0.250

Computing the revised values using the equations started earlier,we get a revised set of values shown as iteration 1.

Using the rank values from Iteration 1 as the new starting values ,we can compute new values for these variables ,shown as Iteration 2. Rank values will continue to change.

Variable	Initial Value	Iteration 1	Iteration 2
Ra	0.250	0.375	0.3125
Rb	0.250	0.125	0.1875
Rc	0.250	0.250	0.250
Rd	0.250	0.250	0.250

Working from values of Iteration2 and so, we can do a few more iterations still the values stabilize. Dataset(10.2) shows the final values after the 8th iteration.

Data Set 10.2

Variable	Initial Value	Iteration 1	Iteration 2	.....	Iteration 8
Ra	0.250	0.375	0.313	.....	0.333
Rb	0.250	0.125	0.188	.....	0.167
Rc	0.250	0.250	0.250	.....	0.250
Rd	0.250	0.250	0.250	.....	0.250

The final rank shows that rank of node A is the highest at 0.333. Thus , the most important node is A . The lowest rank is 0.167 of Rb. Thus , B is the least important node. Nodes C and D are in the middle . In this case,their ranks did not change at all.

The relative scores of the nodes in this network would have been the same irrespective of the initial values chosen for the computations . It may take longer or shorter number of iterations for the results to stabilize for different sets of initial values.

PAGERANK: PageRank is a particular application of the social network analysis techniques above to compute the relative importance of websites in the overall World Wide Web. The data on website and their links is gathered through web crawler bots that traverse through the webpage at frequent intervals. Every webpage is a node in the social network and all the hyperlinks from that page become directed links to other web-pages . Every outbound link from a web-page is considered an outflow of influence of that web-page. An iterative computational technique is applied to compute a relative importance to each page. That score is called PageRank, according to an eponymous algorithm invented by the founders of Google ,the web search company.

PageRank is used by Google for ordering the display of websites in response to search queries. To be shown higher in the search results, many websites owners try to artificially boost their PageRank by creating many dummy websites whose ranks can be made to flow into their desired websites. Also, many websites can be designed to cyclical sets of links from where the web crawler may not be able to break out. These are called **spider traps.**

To overcome these and other challenges , Google includes a Teleporting factor into computing the PageRank. Teleporting assumed that there is a potential link from any node to any other node, irrespective of whether it actually exists. Thus, the influence matrix is multiplied by a weighting factor called Beta with a typical value of 0.85 or 85(percent)%. The remaining weight of 0.15 or 15 (percent)% is given to teleportation . In Teleportation matrix , each cell is given a rank of 1/n , where n is the number of nodes the web. The two matrices are added to compute the final influence matrix. This matrix can be used to iteratively compute the PageRank of all the nodes.

Eighth Semester B.E. Degree Examination,
CBCS - Model Question Paper - 3
BIG DATA ANALYTICS

Time: 3 hrs. Max. Marks: 80

Note : *Answer any FIVE full questions, selecting ONE full question from each module.*

Module - 1

1. a. **Write note on following.** **(08 Marks)**
(1) Rack awareness
(2) HDFS snapshots
(3) HDFS Name-nods Federation

Ans. **(1) Rack Awareness**

Rack awareness deals with data locality. Recall that one of the main design goals of Hadoop MapReduce is to move the computation to the data. Assuming that most data centre networks do not offer bisection bandwidth, a typical Hadoop cluster will exhibit three levels of data locality:

1. Data resides on the local machine (best)
2. Data resides in the same rack (better)
3. Data resides in a different rack (good).

When the YARN scheduler is assigning MapReduce containers to work as mappers, it will try to place the container first on the local machine, then on the same rack, and finally on another rack.

In addition, the NameNode tries to place replicated data blocks on multiple racks for improved fault tolerance. In such a case, an entire rack failure will not cause data loss or stop HDFS from working. Performance may be degraded, however.

HDFS can be made rack-aware by using a user-derived script that enables the master node to map the network topology of the cluster. A default Hadoop installation assumes all the nodes belong to the same (large) rack. In that case, there is no option 3.

(2) HDFS Snapshots

HDFS snapshots are similar to backups, but are created by administrators using the hdfs dfs -snapshot command. HDFS snapshots are read-only point-in-time copies of the file system They offer the following features:

- Snapshots can be taken of a sub-tree of the file system or the entire file system.
- Snapshots can be used for data backup,protection against user errors, and disaster recovery.
- Snapshot creation is instantaneous.
- Blocks on the DataNodes are not copied, because the snapshot files record the block list and the file size. There is no data copying, although it appears tot the user that there are duplicate files.
- Snapshot do not adversely affect regular HDFS operations.

(3) HDFS NameNode Federation

Another important feature of HDFS is NameNode Federation. Older versions of HDFS provided a single namespace for the entire cluster managed by a single NameNode. Thus, the resources of a single NameNode determined the size of the namespace. Federation addresses

this limitation by adding support for multiple NameNode namespaces to the HDFS file system. The key benefits are as follows:

- Namespace scalability. HDFS cluster storage scales horizontally without placing a burden to the NameNode.
- Better performance. Adding more NameNode to the cluster scales the file system read/write operations throughput by separating the total namespaces.
- System isolation. Multiple NameNode enable different categories of applications to be distinguished and users can be isolated to different namespaces.

Figure 1.1 illustrates how HDFS NameNode Federation is accomplished. NameNode1 manges the / research an /marketing namespaces, and NameNode2 manages the / data and / project namespaces. The NameNode do not communicate with each other and the DataNodes "just store data block" as directed by either NameNode.

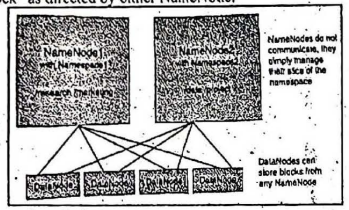


Figure 1.1 IIDFC.NameNode Federation example.

b. **Explain with a concept of running basic Hadoop Benchmarks.** **(8 Marks)**

Ans. **Running Basic Hadoop Benchmarks:** Many Hadoop benchmarks can provide insight into cluster performance. The best benchmarks are always those that reflect real application performance.

The two benchmarks discussed are : **terasort and TestDFSIO,** provide a good sense of how well your Hadoop installation is operating and can be compared with public data published for other Hadoop systems. The results, however, should not be taken as a single indicator for ssytem-wide performance on all applications.

The following benchmarks are designed for full Hadoop cluster installations. These tests assume a multi-disk HDFS environment. Running these benchmarks in the Hortonworks Snadbox or in the pseudo-distributed single-node install is not recommended because all input and output (I/O) are done using a single system disk drive.

Running the Terasort Test

The terasort bechmarks sorts a specified amount of randomly generated data.

This benchmark provides combine testing of the HDFS and MApReduce layers of a Hadoop cluster. A full terasort benchmark run consists of the following three steps:

1. Generating the input data via teragen program.
2. Running the actual terasort benchmark on the input data.
3. Validating the sorted output data via the teravalidate program.

In general, each row is 100 bytes long; thus the total amount of data written is 100 times the number of rows specified as part of the benchmark (i.e., to write 100GB of data, use 1 billion rows). The input and output directories need to be specified in HDFS. The following sequence of commands will run the benchmark for 50GB of data as user hdfs. Make sure the /user/hdfs directory exists in HDFS before running the benchmarks.

1. Run teragen to generate rows of random data to sort.

$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-examples.jar teragen 500000000
→/user/hdfs/TeraGen-50GB

2. Run terasort to sort the database.

$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-examples.jar terasort
→/user/hdfs/TeraGen-50GB /user/hdfs/TeraSort-50GB

3. Run teravalidate to validate the sort.

$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-example.jar teravalidate
→/user/hdfs/TeraSort-50GB /user.hdfs/TeraValid-50GB

To report results, the time for the actual sort (terasort) is measured and the benchmark rate in megabytes/second (MB/s) is calculated. For best performance, the actual terasort benchmark should be run with a replication factor of 1. In addition, the default number of terasort reducer tasks is set to 1. Increasing the number of reducers often helps with benchmark performance. For example, the following command will instruct terasort to use four reducer tasks:

$.yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-example.jar terasort
→ -Dmapred.reduce.tasks=4 /user/hdfs/TeraGen-50GB /user/hdfs/TeraSort-50GB Also, do not forget to clean up the terasort data between runs (and after testing is finished). The following command will perform the cleanup for the previous example:

$ hdfs dfs -rm -r -skipTrash Tera*

Running the TestDFSIO Benchmark

Hadoop also includes an HDFS benchmark application called testDFSIO. The TestDFSIO benchmark is a read and write test for HDFS. That is, it will write or read a number of files to and from HDFS and is designed in such a way that it will use one map task per file. The file size and number of files are specified as command-line arguments. Similar to the terasort benchmark, you should run this test as user hdfs. Similar to terasort, TestDFSIO has several steps. In the following example.

16files of size1GB are specified. Note that the TestDFSIO benchmark is part of the hadoop-mapreduce-client-jobclient.jar. Other benchmarks are also available as part of this jar file. Running it with no arguments will yield a list. In addition to TestDFSIO, NNBench (load testing the NameNode) and MRBench (load testing the MapReduce framework) are commonly used Hadoop benchmarks. Nevertheless, TestDFSIO is perhaps the most widely reported of these benchmarks. The steps to run TestDFSIO are as follows:

1. Run TestDFSIO in write mode and create data.

$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-client-jobclient-tests.jar
→ TestDFSIO -write -nrFiles 16 -fileSize 1000

Example results are as follows (data and time prefix removed).

fs.TestDFSIO: -----TestDFSIO -----: write

fs.TestDFSIO: Date & time: Thu May 14 10:39:33 EDT 2015 fs.TestDFSIO: Number of files: 16

fs.TestDFSIO: Total MBytes processed: 16000.0 fs.TestDFSIO: Throuhput mb/sec: 14.890106361891005 fs.TestDFSIO: Average IO rate mb/sec: 15.690713882446289 fs.TestDFSIO: IO rate std deviation: 4.022703520|665595 fs.TestDFSIO: Test exec time sec: 105.631

2. Run TestDFSIO in read mode.

$.yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-client-jobclient-tests.jar
→ TestDFSIO -read -nrFiles 16 -fileSize 1000

Example results are as follows (data and time prefix removed). The large standard deviation is due to the placement of tasks in the cluster on a small four-node cluster. fs.TestDFSIO: -------TestDFSIO ----- : read
fs.TestDFSIO: Data & time: Thu May 14 10:44:09 EDT 2015 fs.TestDFSIO: Number of files: 16
fs.TestDFSIO: Total MBytes processed: 16000.0 fs.TestDFSIO: Throughput mb/sec: 32.38643494172466 fs.TestDFSIO: Average IO rate mb/sec: 58.72880554199219 fs.TestDFSIO: IO rate std deviation: 64.60017624360337 fs.TestDFSIO: Test exec time sec: 62.798

3. Clean up the TestDFSIO data.

$ yarn jar $HADOOP_EXAMPLES/hadoop-mapreduce-client-jobclient-tests.jar
→ TestDFSIO -clean

Running the TestDFSIO and terasort benchmarks help you gain confidence in a Hadoop installation and detect any potential problems. It is also instructive to view the Amabri deshboard and the YARN web GUI (as described previously) as the tests run.

Managing Hadoop MapReduce Jobs

Hadoop MapReduce jobs can be managed using the mapred job command. The most important options for this command in terms of the examples and benchmarks are **-list, -kill, and -status.**

In particular, if you need to kill one of the examples or benchmarks, you can use the mapred job -list command to find the job -id and then use mapred job -kill <job-id> to kill the job across the cluster. MapReduce jobs can also be controlled at the application level with the yarn application command. The possible options for mapred job are as follows:

$ mapred job
Usage: CLI <command> <args> [-submit <job-file>]
[-status <job-id>]
[-counter <job-id> <group-name> <counter-name>] [-kill <job-id>]
[-set-priority <job-id> <priority>] . Valid values for priorities are: VERY_HIGH HIGH NORMAL LOW VERY_LOW
[-events <job-id> <from-event-#> <#-of-events>] [-history <jobHistoryFile>]
[-list [all]]
[-list-active-trackers]
[-list-blacklisted-trackers]
[-list-attempt-ids <job-id> <task-type> <task-state>] . Valid values for <task-type> are REDUCE MAP. Valid values for <taks-state> are running, completed
[-kill-task <task-attempt-id>] [-fail-task <task attempt-id>]
[-logs <job-id> <task-attempt-id>]
Generic options supported are
-conf <configuration file> specify an application configuration file
-D <property=values> use value for given property
-fs <local | namenode:port> specify a namenode
-jt <local | resourcemanager:port> specify a ResourceManager
-files <comma separated list of files> specify comma separated files to be copied to the map reduce cluster
-libjars <comma separated list of jars> specify comma separated jar files to include in the classpath.
-archives <comma separated list of archives> specify comma separated

archives to be unarchived on the compute machines.
The general command line syntax is
bin/hadoop command (genericOptionas) [commandOptions]

OR

2. a. Write a short note on following
i. Speeuulative execution
ii. Hadoop Map reduce hardware. **(4 Marks)**

Ans. **i. Speculative Execution:** One of the challenges with many large cluster is the inability to predict or manage unexpected system bottlenecks or failures. In theory, it is possible to control and monitor resources so that network traffic and processor load can be evenly balanced; in practice, however, this problem represents a difficult challenge for large systems. Thus, it is possible that a congested network, slow disk controller, failing disk, high processor load, or some other similar problem might lead to slow performance without anyone noticing.

When one part of a MapReduce process runs slowly, it ultimately slows down everything else because the application cannot complete until all processes are finished. The nature of the parallel MapReduce model provides an interesting solution to this problem. Recall that input data are immutable in the MapReduce process. Therefore,it is possible to start a copy of a running map process without disturbing any other running mapper processes. For example, suppose that as most of the map tasks are coming to a close, the ApplicationMaster that some are still running and schedules redundant copies of the remaining jobs on less busy or free servers. Should the secondary processes finish first, the other first processes are then terminated (or vice versa). This process is known as speculative execution. The same approach can be applied to reducer processes that seem to be taking a long time. Speculative execution can seem to have a slow spot. It can also be turned off and on in the mapred-site .xml configuration file.

ii. Hadoop MapReduce Hardware

The capability of Hadoop MapReduce and HDFS to tolerate server-or even whole rack-failures can influence hardware designs. The use of commodity (typically x86_64) servers for Hadoop clusters has made low-cost, high-cost, high-availability implementations of Hadoop possible for many data centres. Indeed, the Apache Hadoop philosophy seems to assume on a cluster.

The use of server nodes for both storage (HDFS) and processing (mappers, reducers) is somewhat different from the traditional separation of these two tasks in the data centre. It is possible to build Hadoop system and separate the roles (discrete storage and processing nodes). However,a majority of Hadoop systems use the general approach where servers enact both roles. Another interesting feature of dynamic MapReduce execution is the capability to tolerate dissimilar servers. That is, old and new hardware can be used together. Of course, large disparities in performance will limit the faster systems, but the dynamic nature of MapReduce execution will still work effectively on such systems.

b. Explain compiling and running the Hadoop Grep chaining example with program. **(08 Marks)**

Ans. The Hadoop Grep.java example extracts matching string from text files and counts how many times they occurred. The command works differently from the *nix grep command in that it does not display the complete matching line, only the matching string. If matching lines are needed for the string foo, use .*foo.* as a regular expression.

The program runs two map / reduce jobs in sequence ans is an example of MapReduce

chaining. The first job counts how many times a matching string occurs in the input, and the second job sorts matching strings by their frequency and stores the output in a single file. Listing 1.1 displays the source code for Grep.java.

Note that all the Hadoop example source files can be extracted by locating the hadoop-mapreduce-example-*-sources.jar either from a Hadoop distribution or from the Apache Hadoop website (as part of a full Hadoop package) and then extracting the files using the following command (your version tag may be different):

$ jar xf hadoop-mapreduce-example-2.6.0 sources.jar

Listing 1.1 Hadoop Grep.java Example

```
package org.apache.hadoop.examples;
import java.util.Random;
import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.conf.
Configuration; import org.apache.hadoop.fs.Filesystem; import org.apache.hadoop.fs.path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.SequenceFileInputFormat;
import org.apache.hadoop.mapreduce.lib.map.InverseMapper
import org.apache.hadoop.mapreduce.lib.map.RegexMapper;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.SequenceFileOutputFormat;
import org.apache.hadoop.mapreduce.lib.reduce.LongSumReducer;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
/*Extracts matching regexs from input files and counts them. */
public class Grep extends configured implements Tool {
private Grep ( ) { ) // singleton.
public int run (String [ ] args) throws Exception { if (args .length <3) {
System.out.prit ln ("Grep <inDri> <outDir> <regex> [<group>]");
return 2;
}
Path tempDir =
new path ("grep-temp-"+
Integer .toString (new Random ( ) .
nextInt (Integer .MAX_VALUE) ) ;
Configuration conf=getCont ( ) ;
conf.set (RegexMapper .PATTERN, arge[2] );
if (args. length = = 4)
conf .set (RegexMapper .GROUP, args [3] );
Job grepJob = new Job (counf) ;
try {
grepJob.setJobName ("grep-search");
FileInputFormat.setInputPaths (grepJob, args[0]);
grepJob.setMapperClass (RegexMapper.class) ;
grepJob.setCombinerClass (LongSumReducer.class);
grepJob.setReducerClass (LongSumReducer.class);
FileOutputFormat.setOutputPath (grepJob, tempDir);
grepJob.setOutputFormatClass (SequenceFileOutputFormat.class);
```



```
grepJob.setOutputKeyClass (Text.class);
grepJob.setOutputValuesclass (LongWritable.class);
grepJob.waitForCompletion (true);
Job sortJob = new Job (conf); sortJob.setJobName ("grep-sort");
FileInputFormat.setInputPaths (sortJob, tempDir);
sortJob.setInputFormatClass(SequenceFileInputFormat.class);
sortJob.setMapperClass(InverseMapper.class);
sortJob.setSortNumReduceTasks (1); //Write a single file FileOutputFormat.
setOutputPath(sortJob,new path(args [1]); . LongWritable.DecreasingComparator.class);
sortJob.waitForCompletion (true);
}
finally {
FileSystem.get(conf).delete (tempDir, true);
}
return 0;
}
public static void main (String [ ] args) throws Exception {
int res = ToolRunner.run(new Configuration ( ), new Grew ( ) , args);
System.exit(res);
}
}
```

In the preceding code, each mapper of the first job takes a line as input and matches the user-provided regular expression against the line. The RegexMapper class is used to perform this task and extracts text matching using the given regular expression. The matching strings are output as <matching string, 1> pairs. As in the previous WordCount example, each reducer sum up the number of matching strings and employs a combiner to do local sums. The actual reducer uses the LongSumReducer class that outputs the sum of long values per reducer input key.

The second job takes the output of the first job as its input. The mapper is an inversemap that reverses (or swaps) its input <key, value> pairs into <value,key>.There is no reduction step, so the IdentityReducer class is used by default. All input is simply passed to the output (Note: There is also an IdentityMapper class.) The number of reducers is set to 1, so the output is stored in one file and it is sorted by count in descending order. The output text file contains a count and a string per line.

The example also demonstrates how to pass a command-line parameter to a mapper or a reducer.

The following discussion describes how to compile and run the Grep.java example. The steps are similar to the previous WordCount example:

1. Create a directory for the application classes as follows:

$ mkdir Grep_classes

2. Compile the WordCount.java program using the following line:

$ javac -cp `hadoop classpath` -d Grep_classes Grep.java

3. Create a Java archive using the following command:

$ jar -cvf Grep.jar -C grep_classes/.

If needed, create a directory and move the war-and-peace.txt file into HDFS:

$ hdfs dfs -mkdir war-and-peace-input

$ hdfs dfs -put war-and-peace.txt war-and-peace-output

As always, make sure the output directory has been removed by issuing the following command:

```
$ hdfs dfs -rm -r -skipTrash war-and-peace-output
```

Entering the following command will run the Grep program:

```
$ hadoop jar Grep.jar org.apache.hadoop.example.Grep war-and-peace-input
→ war-and-peace-output Kutuzov
```

As the example runs, two stages will be evident. Each stage is easily recognizable in the program output. The results can be found by examining the resultant output file.

```
$ hdfs dfs -cat war-and-peace-output/part-r-00000
530     Kutuzov
```

c. **Explain command line log voiding.** **(04 Marks)**

Ans. MapReduce logs can also be viewed from the command line. The yarn logs command enables the logs to be easily viewed together without having to hunt for individual log files on the cluster nodes. As before, log aggregation is required for use. The options to yarn logs are as follows:

```
$ yarn logs
```

Retrieve logs for completed YARN applications.

```
usage: yarn logs -applicationId <application ID> [OPTIONS]
general options are:
-appOwner <Application Owner>  AppOwner (assumed to be current user if not
                               specified)
-containerId <Container ID>ContainerId (must be specified if node address is
                               specified)
-nodeAddress <Node Address>    NodeAddress in the format nodename:
                               port (must be specified if container id is specified)
```

For example, after running the pi example program (discussed in Chapter 4), the logs can be examined as follows:

```
$ hadoop jar $HADOOP_EXAMPLES/hadoop-mapreduce-examples.jar pi 16
100000
```

After the pi example completes, note the applicationId, which can be found either from the application output or by using the yarn application command. The applicationId will start with application_and appear under the Application-Id column.

```
$ yarn application -list -appStates FINISHED
```

Next, run the following command to produce a dump of all the logs for that application. Node that the output can be long and is best saved to a file.

```
$ yarn logs -applicationId application_1432667013445_0001 > AppOut
```

The AppOut file can be inspected using a text editor. Note that for each container, stdout, stderr, and syslog are provided . The list of actual containers can be found by using the following command:

```
$ grep -B 1 ===== AppOut For example (output truncated):
[...]
Container: container_1432667013445_0001_01_000008 on limulus_45454
==========================================
--
Container: container_1432667013445_0001_01_000010 on limulus_45454
==========================================
--
```



```
Container: container_1432667013445_0001_01_000001 on n0_45454
==========================================
--
Container: container_1432667013445_0001_01_000023 on n1_45454
==========================================
[...]
```

A specific container can be examined by the containerId and the nodeAddress from the preceding output. For example, container_1432667013445_0001_01_000023 can be examined by entering the command following this paragraph. Note that the node name (n1) and port number are written as n1_45454 in the command output. To get the nodeAddress, simply replace the _ with a: (i.e., -nodeAddress n1:45454). Thus, the results for a single container can be found by entering this line:

```
$ yarn logs -applicationId application_1432667013445_0001_containerId
→ container_1432667013445_0001_01_000023 -nodeAddress n1:45454 | more
```

Module -2

3. a. **Explain with an diagrams DAG work flows?** **(10 Marks)**

Ans. Oozie is a workflow director system designed to run and manage multiple related Apache Hadoop jobs. For instance, complete data input and analysis may required several discrete Hadoop jobs to be run as a workflow in which the output of one job serves as the input for a successive job. Ooze is not a substitute for the YARN scheduler. That is, YARN manages resources for individual Hadoop jobs, and Oozie provides a way to connect and control Hadoop jobs on the cluster.

Oozie workflow jobs are represented as directed acyclic graphs (DAGs) of actions. (DAGGs are basically graphs that cannot have directed loops.) Three types of Oozie jobs are permitted:

- Workflow-a specified sequence of Hadoop jobs with outcome-based decision points and control dependency. Progress from one action to another cannot happen until the first action is complete.
- Coordinate-a scheduled workflow job that can run at various time intervals or when data become available.
- Bundle-a higher-level Oozie abstraction that will batch a set of coordinator jobs. Oozie is integrated with the rest of the Hadoop stack, supporting several types of Hadoop jobs out of the box (e.g., Java MapReduce, Streaming MapReduce, Pig, Hive, also Sqoop) as well as system-specific jobs (e.g., Java program and shell scripts). Oozie also provides a CLI and a web UI for monitoring jobs.

Figure 3.1 depicts a simple Oozie workflow. In this case, Oozie runs a basic MapReduce operation. If the application was successful, the job end; if an error occurred, the job is killed. Oozie workflow definitions are written in hPDL (an XML Process Definition Language). Such workflow contain several types of nodes:

- Control flow nodes define the beginning and the end of a workflow. They include start, end, and optional fail nodes.
- Action nodes are where the actual processing taks are defined. When an action node finishes, the remote systems notify Oozie and the next node in the workflow is executed. Action nodes can also include HDFS commands.
- Fork/join node enable parallel execution of tasks in the workflow. The fork node enable two or more tasks to run at the same time. A join node represents a redezvous point that must wait until all forked tasks complete.

- Control flow nodes enable decisions to made about the previous task. Control decisions are based on the results of the previous action (e.g., file size or file existence). Decision nodes are essentially switch-case statements that use JSP

EL (Java Server pages-Expression Language) that evaluate to either true or false. Figure 3.2 depicts a more complex workfolw that uses all of these node types.

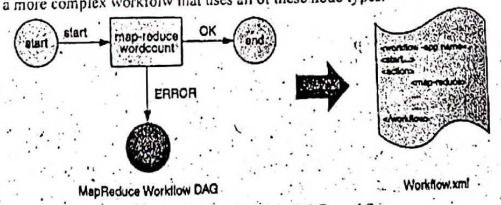


Figure 3.1 A simple Oozie DAG workflow

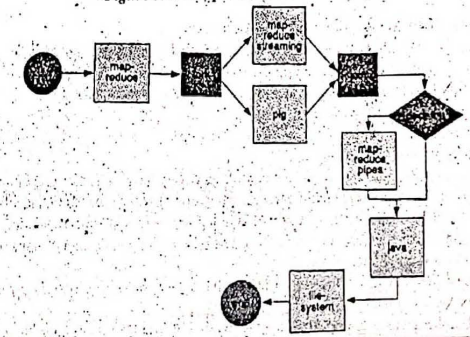


Fig 3.2 A more complex Oozie DAG workflow

b. Explain the following in the Application frame work? (06 Marks)

(i) Apache Tez

(ii) Apache Giraph

(iii) Hamster Hadoop and MPI on the same cluster.

(iv) Apachespark.

Ans. (i) Apache Tez

One great example of a new YARN framework is Apache Tez. Many Hadoop jobs involves the execution of a complex directed acyclic graph (DAG) of task using separate MapReduce stage. Apache Tez generalizers this process and enables these tasks be spread across stages so that they can be run as a single, all-encompassing job. Tez can be used as a MapReduce replacement for projects such as Apache Hive and Apache Pig. No changes are needed to the Hive or Pig applications.

ii) Apache Giraph

Apache Giraph is an iterative graph processing system built for high scalability. Facebook, twitter, and LinkedIn use it to create social graphs users. Giraph was originally written to

run on standard Hadoop V1 using the MapReduce framework, but that approach proved inefficient and totally unnatural for various reasons. The native Giraph implementation under YARN provides the user with an iterative processing model that is not directly available with MapReduce. Support for YARN has been present in Giraph since its own version 1.0 release. In addition, using the flexibility of YARN, the Giraph developers plan on implementing their own web interface to monitor job progress

iii) Hamster: Hadoop and MPI on the Same Cluster

The Message Passing Interface (MPI) is widely used in high-performance computing (HPC). MPI is primarily a set of optimized message-passing library calls for C, C++, and Fortran that operate over popular server interconnects such as Ethernet and InfiniBand. Because users have full control over their YARN containers, there is no reason why MPI applications cannot run within a Hadoop cluster. The Hamster effort is a work-in-progress that provides a good discussion of the issues involved in mapping MPI to a YARN cluster. Currently, an alpha version of MPICH2 is available for YARN that can be used to run MPI applications.

iv)Apache Spark

Spark was initially developed for applications in which keeping data in memory improves performance, such as iterative algorithms, which are common in machine learning, and interactive data mining. Spark differs from classic MapReduce in two important ways. First, Spark holds intermediate results in memory, rather than writing them to disk. Second, Spark holds supports more than just MapReduce functions; that is, it greatly expands the set of possible analyses that can be executed over HDFS data stores. It also provides APIs in Scala, Java, and Python.

Since 2013, Spark has been running on production YARN clusters at Yahoo!. The advantage of porting and running Spark on top of YARN is the common resource management and a single underlying file system.

OR

4. a. How to change Hadoop properties? (12 Marks)

Ans. One of the challenges of managing a Hadoop cluster is managing changes to cluster wide configuration properties. In addition to modifying a large number of properties making changes to a property often required daemons (and dependent daemons0 across the entire cluster. This process is tedious and time consuming. Fortunately, Ambari provides an easy way to manage this process.

Each service provides a Configs tab that opens a form displaying all the possible service properties. Any service property can be change (or added) using this interface. As an example, the configuration properties for the YARN scheduler as shown in Figure 4.1.

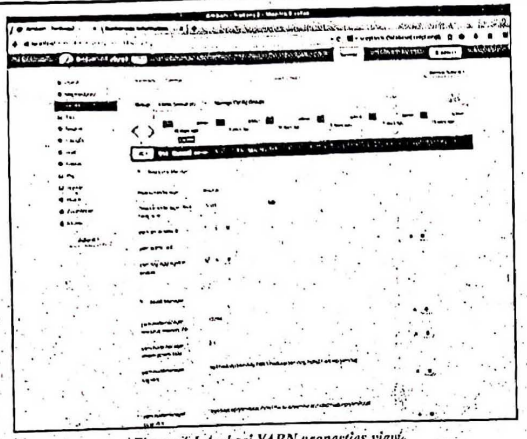


Figure 4.1 Ambari YARN properties view

The number can be options offered depends on the service; the full range of YARN properties can be viewed by scrolling down the form. To easily view the application logs, this property must be set to true. This property is normally on by default. As an example for our purpose here, we will use the Ambari interface to disable this feature. As shown in Figure 4.2, when a property is changed, the green Save button becomes activated.

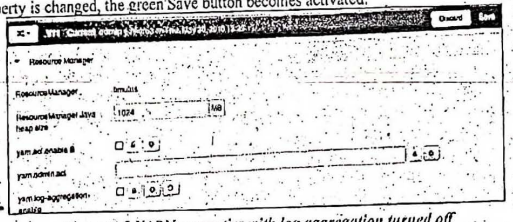


Figure 4.2 YARN properties with log aggregation turned off

Changes do not become permanent until the user clicks the Save button. A save / notes window will then be displayed. It is highly recommended that historical notes concerning the change be added to this window.

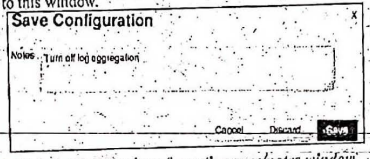


Figure 4.3 Ambari configuration save/notes window.

Once the user adds any notes add clicks the Save button, another window, shown in Figure 4.4, is presented. This window confirms that the properties have been saved. Once the new property is changed, an orange Restart button will appear at the top left of the window. The new property will not take effect until the required services are restarted. As shown in Figure 4.5, the Restart button provides two options: Restart All and Restart NodeManagers. To be safe, the Restart All should be used. Note that Restart All does not mean all the Hadoop service will be restarted ; rather, only those that use the new property will be restarted.

After the user clicks Restart All, a confirmation window, shown in Figure 4.6 will be displayed. Click Confirm Restart All to begin the cluster-wide restart.

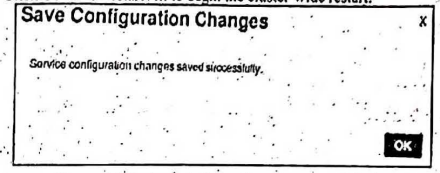


Figure 4.4 Ambari configuration change notification

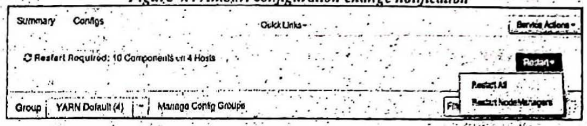


Figure 4.5 Ambari Restart function appears after changes in service properties

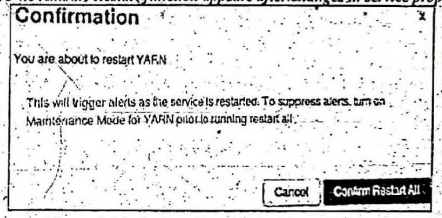


Figure 4.6 Ambari confirmation box for service restart

Similar to the DataNode restart example, a progress window will be displayed. Again, the progress bar is for the entire YARN restart. Details from the logs can be found by clicking the arrow to the right of the bar (see Figure 4.7).

Once the restart is complete, run a simple example and attempt to view the logs using the YARN ResourceManager Application UI. (You can access the UI from the Quick Links pull-down menu in the middle of the YARN series window.) A message similar to that in Figure 4.8 will be displayed .

Ambari tracks all changes made to system properties. As can be seen in Figure 4.1 and in more detail in Figure 4.9, each time a configuration is changed, a new version is created. Reverting back to a previous version results in a new version. You can reduce the potential for version confusion by providing meaningful comments for each change (e.g., Figure 4.3 and Figure 4.11). In the preceding example, we created version 12 (V12). The

current version is indicated by a green Current label in the horizontal version boxes or in the dark horizontal bar. Scrolling though the version boxes

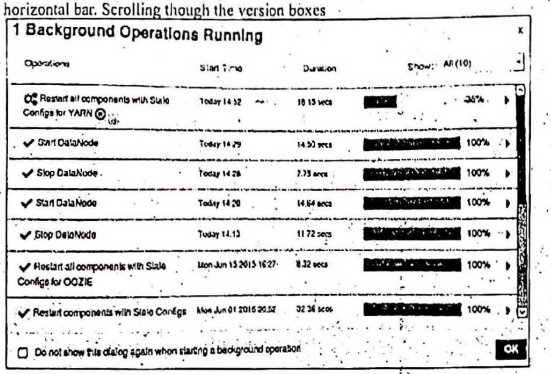


Figure 4.7 Ambari progress window for cluster wide YARN start

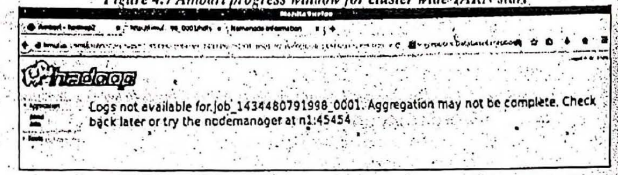


Figure 4.8 YARN ResourceManager interface with log aggregation turned off

or pulling down the menu on the left-hand side of the dark horizontal bar will display the previous configuration versions.

To revert to a previous version, simply select the version from the version boxes or the pull-down menu. In Figure 4.10, the user has selected the previous version by clicking the Make Current button in the information box. This configuration will return to the previous state where log aggregation is enabled.

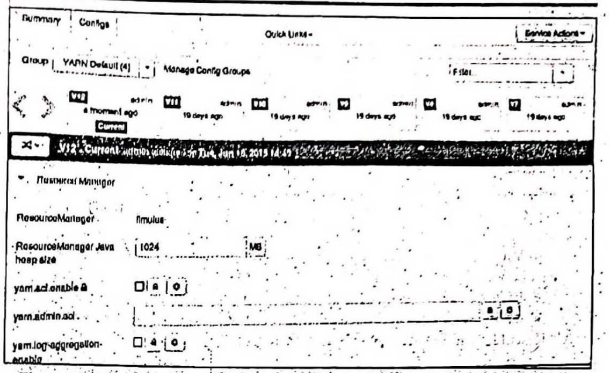


Figure 4.9 Ambari configuration change management for YARN service (Version V12 current)

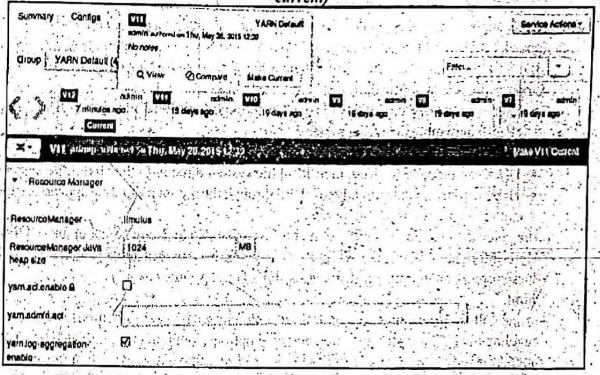


Figure 4.10 Reverting to previous YARN configuration (V11) with Ambari

As shown in Figure 4.11, a confirmation / notes window open before the new configuration is saved. Again, it is suggested that you provide note about the change in the Notes text box. When the save step is complete, the Make Current button will restore the previous configuration. The orange Restart button will appear and indicate that a service restart is needed before the changes take effect.

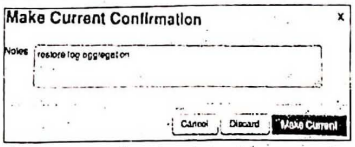


Figure 4.11 Ambari confirmation window for a new configuration

There are several important points to remember about the Amabri versioning tool:

- Every time you change the configuration, a new version is created. Reverting to a previous version created a new version.
- You can view or compare a version to other versions without having to change or restart service. (See the buttons in the V11 box in Figure 4.10.)
- Each service has its own version record.
- Every time you change the properties, you must restart the service by using the Restart button. When in doubt, restart all services.

b. Define the capabilities and configuration steps of an NFSV3 Gateway to HDFS **(04 Marks)**

Ans. Configuring an NFSv3 Gateway to HDFS

HDFS supports an NFS version 3 (NFSv3) gateway. This feature enables files to be easily moved between HDFS and client systems. The NFS gateway supports NFSv3 and allows HDFS to be mounted as part of the client's local file system. Currently the NFSv3 gateway supports the following capabilities:

- Users can browse the HDFS file system through their local file system using an NFSv3 client-compatible operating system.
- Users can download files from the HDFS file system to their local file system.
- Users can upload files from their local file system directly to the HDFS file system.
- Users can stream data directly to HDFS through the mount point. File append is supported, but random write is not supported.

The gateway must be run on the same host as a DataNode, NameNode, or any HDFS client. More information about the NFSv3 gateway can be found at https:/' hadoop. apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsNfsGateway .html.

In the following example, a simple four-node cluster is used to demonstrate tne steps for enabling the NFSv3 gateway. Other potential options, including those, related to security, are not addressed in this example. A DataNode is used as the gateway node in this example, and HDFS is mounted on the main (login) cluster node.

Step 1: Set Configuration Files

Several Hadoop configuration files need to be changed. In this example, the Ambari GUI will be used to alter the HDFS configuration files. Do not save the changes or restart HDFS until all the following changes are made. If you are not using Ambari, you must change these files by hand and then restart the appropriate services across the cluster. The following environmen is assumed:

- OS: Linux
- Platform: RHEL 6.6
- Hortonworks HDP 2.2 with Hadoop version: 2.6

Several properties need to be added to the /etc/hadoop/conf ig/core-site.xml file. Using

Ambari, go to the HDFS service window and select the Configs tab. Toward the bottom of the screen, select the Add Property link in the Custom core- site.xml section. Add the following two properties (the item used for the key field in Ambari is the name field included in this code):

```
<property>
<name>hadoop.proxyuser.root.groups</name>
<value>*</value>
</property>
<property>
<name>hadoop.proxyuser.root.hosts</name>
<value>*</value> </property>
```

The name of the user who will start the Hadoop NFSv3 gateway is placed in the name field. In the previous example, root is used for this purpose. This setting can be any user who starts the gateway. If, for instance, user nf sadmin starts the gateway, then the two names would be hadoop.proxyuser.nfsadmin.groups and hadoop.proxyuser. nfsadmin.hosts. The * value, entered in the preceding lines, opens the gateway to all groups and allows it to run on any host. Access is restricted b entering groups (comma separated) in the group's property. Entering a host name for the host's property can restrict the host running the gateway.

Next, move to the Advanced hdfs-site.xml section and set the following property: property>

```
<name>dfs.nfs3.dump.dir</name>
<value>/tmp/.hdfs-nfs</value>
</property>
```

The NFSv3 dump directory is needed because the NFS client often recorders writes. Sequential writes can arrive at the NFS gateway in random order. This directory is IZ to temporarily save out-of-order writes before writing to HDFS. Make sure the dump directory has enough space. For example, if the application uploads 10 files, each of size 100MB, it is recommended that this directory have 1GB of space to cover a worst-case write reorder for every file.

Once all the changes have been made, click the green Save button and note the changes you made to the Notes box in the Save confirmation dialog. Then restart all of HDFS by clicking the orange Restart button.

Step 2: Start the Gateway

Log into a DataNode and make sure all NFS services are stopped. In this example, Data Node nO is used as the gateway.

```
# service rpebind stop # service nfs stop
```

Next, start the HDFS gateway by using the hadoop-daemon script to start portmap and nfs3 as follows :

```
#/usr/hdp/2.2.4.2-2/hadoop/sbin/hadoop-daemon.sh start portmap #/usr/hdp/2.24.2-2/hadoop.sbin/hadoop-daemon.sh start nfs3
```

The portmap daemon will write its log to /vvar /log/hadoop/root/hadoop-root-nfs3-n0.log

To confirm the gateway is working, issue the following command. The output should look like the following :

```
#rpcinfo -p n0
```

program	vers	proto	port	service
100005	2	top	4242	mountd

100000	2	udp	111	portmapper
100000	2	top	111	portmapper
100005	1	tcp	4242	mounted
100003	3	tcp	2049	nfs
100005	1	udp	4242	mounted
100005	3	udp	4242	mounted
100005	3	tcp	4242	mounted
100005	2	udp	4242	mounted

Finally, make sure the mount is available by issuing the following command :
#showmount -e n0
Export list for n0 :
/*
If the rpcinfo or showmount command does not work correctly, check the previously mentioned of files for problems.

Step 3 : Mount HDFS

The final step is to mount HDFS on a client node. In this example, the main login node is used. To mount the HDFS files, exit from the gateway node and create the following directory :
#mkdir /mnt / hdfs
The mount command is as follows. Note that the name of the gateway node will be different on other clusters, and an IP address can be used instead of the node name. #mount -t nfs -0 vers = 3, proto = tcp, nolock n0 : / /mnt / hdfs /
Once the file system is mounted, the files will be visible to the client users. The following command will list the mounted file system:
is/mnt/hdfs
app-logs apps benchmarks hdp mapred mr-history system tmp user var The gateway in the current Hadoop release uses AUTH_UNIX-style authentication an requires that the login user name on the client march the user name that NFS passes to HDFS. For example, if the NFS client is user admin, the NFS gateway will access HDFS as user admin and existing HDFS permissions will prevail.
The system administrator must ensure that the user on the NFS client machine has the same user name and user ID as that on the NFS gateway machine. This is usually not a problem if you use the same user management system, such as LDAP/NIS, to create and deploy users to cluster nodes.

Module -3

5. a. **Describe list of business intelligence tools used in the organization. Explain any 2 of them used in your organization.** **(10 Marks)**

Ans. According to the list of best business intelligence tools prepared by experts from Finances Online the leading solutions in this category comprise of systems designed to capture, categorize, and analyze corporate data and extract best practices for improved decision making. The more advanced the system is, the more data sources it will combine, including internal metrics coming from different company departments, and external data extracted from third-party systems, social media channels, emails, or even macroeconomic data. Ultimately, business intelligence software helps companies gain insight on their overall growth, sales trends, and customer behavior.

1. Sisense	20. Palo OLAP Server
2. Actuate Business Intelligence and Reporting Tools (BIRT)	21. Pentaho
3. icCube	22. Profit base
4. Domo	23. QlikView
5. Board Management Intelligence Toolkit	24. Rapid insight.
6. Clear Analytics	25. SAP business intelligence
7. Ducen	26. SAP BusinessObjects
8. Gooddata	27. SAP NetWeaver BW
9. IBM Cognos Intelligence	28. SAS BI
10. Insightsquared	29. Silvon
11. JasperSoft	30. Solver
12. Looker	31. SpagoBI
13. Microsoft BI platform	32. SQL Server Analysis Services
14. MicroStrategy	33. Style Intelligence
15. MITS	34. Syntell solutions
16. Openl	35. Targit
17. Oracle BI	36. Vismatica
18. Oracle Enterprise BI Server	37. WebFOCUS
19. Oracle Hyperion System	38. Yellowfin BI

The BI tool used in our organization : Education
As higher education becomes more expensive and competitive, it is a great user of data-based decision-making. There is a strong need for efficiency, increasing revenue, and improving the quality of student experience at all levels of education.
1. Student enrolment (recruitment and retention): Marketing to new potential students requires schools to develop profiles of the students that are most likely to attend. Schools can develop models of what kinds of students are attracted to the school, and then reach out to those students. The students at risk of not returning can be flagged, and corrective measures can be taken in time.
2. Course offerings: Schools can use the class enrolment data to develop models of which new courses are likely to be more popular with students. This can help increase class size, reduce costs, and improve student satisfaction.
3. Alumni pledges: Schools can develop predictive models of which alumni are most likely to pledge financial support to the school. Schools can create a profile for alumni more likely to pledge donations to the school. This could lead to a reduction in the cost of mailings and other forms of outreach to alumni.

b. **What are the sources and types of data for a data warehouse? How will data warehousing evolve in the age of social media?** **(06 Marks)**

Ans. **Data Sources:** DWs are created from structured data sources. Unstructured data, such as text data, would need to be structured before inserted into DW.
1. Operations data include data from all business applications, including from ERPs systems that form the backbone of an organization's IT systems. The data to be extracted will depend upon the subject matter of DW. For example, for a sales/marketing DW, only the data about customers, orders, customer service, and so on would be extracted.

2. Other applications, such as point-of-sale (POS) terminals and e- commerce applications, provide customer-facing data. Supplier data could come from supply chain management systems. Planning and budget data should also be added as needed for making comparisons against targets.
3. External syndicated data, such as weather or economic activity data, could also be added to DW, as needed, to provide good contextual information to decision makers.

Three main types of Data Warehouses are:

1. Enterprise Data Warehouse:

Enterprise Data Warehouse is a centralized warehouse. It provides decision support service across the enterprise. It offers a unified approach for organizing and representing data. It also provide the ability to classify data according to the subject and give access according to those divisions.

2. Operational Data Store:

Operational Data Store, which is also called ODS, are nothing but data store required when neither Data warehouse nor OLTP systems support organizations reporting needs. In ODS, Data warehouse is refreshed in real time. Hence, it is widely preferred for routine activities like storing records of the Employees.

3. Data Mart:

A data mart is a subset of the data warehouse. It specially designed for a particular line of business, such as sales, finance, sales or finance. In an independent data mart, data can collect directly from sources.

A DW project reflects a significant investment into IT. All of the best practices in implementing any IT project should be followed.

1. The DW project should align with the corporate strategy. Top management should be consulted for-setting objectives. Financial viability Return on Investment (ROI) should be established. The project must be managed by both IT and business professionals. The DW design should be carefully tested before beginning development work. It is often much more expensive to redesign after development work has begun.

2. It is important to manage user expectations. DW should be built incrementally. Users should be trained in using the system, and absorb the many features of the system.

3. Quality and adaptability should be built in from the start. Only cleansed and high-quality data should be loaded. The system should be able to adapt to new access tools. As business needs change, new data marts can be created for new needs

OR

6. a. **Why is data preparation so important and time consuming?** **(04 Marks)**

Ans. Data cleansing and preparation is a laborintensive or semiautomated activity that can take up to 60 to 70 percent of the time needed for a data mining project.

1. Duplicate data needs to be removed. The same data may be received from multiple sources. When merging the data sets, data must be de-duped.
2. Missing values need to be filled in, or those rows should be removed from analysis. Missing values can be filled in with average or modal or default values.
3. Data elements may need to be transformed from one unit to another. For example, total costs of health care and the total number of patients may need to be reduced to cost/patient to allow comparability of that value.
4. Continuous values may need to be binned into a few buckets to help with some analyses. For example, work experience could be binned as low, medium, and high.
5. Data elements may need to be adjusted to make them comparable over time. For example, currency values may need to be adjusted for inflation; they would need to be converted to the same base year for comparability. They may need to be converted to a common currency.
6. Outlier data elements need to be removed after careful review, to avoid the skewing of results. For example, one big donor could skew the analysis of alumni donors in an educational setting.
7. Any biases in the selection of data should be corrected to ensure the data is representative of the phenomena under analysis. If the data includes many more members of one gender than is typical of the population of interest, then adjustments need to be applied to the data.
8. Data should be brought to the same granularity to ensure comparability. Sales data may be available daily, but the sales person compensation data may only be available monthly. To relate these variables, the data must be brought to the lowest common denominator, in this case, monthly.
9. Data may need to be selected to increase information density. Some data may not show much variability, because it was not properly recorded or for any other reasons. This data may dull the effects of other differences in the data and should be removed to improve the information density of the data.

b. **Describe some key steps in data visualization.** **(08 Marks)**

Ans. Data has been described as the new raw material for business and the "oil of the 21st century. The volume of data used in business, research and technological development is massive and continues to grow. For instance at Elsevier, there are about 700 million articles per year downloaded from ScienceDirect, 80,000 institution profiles on Scopus, 13 million researcher profiles on Scopus and 3 million researcher profiles on Mendeley. It becomes harder and harder for a user to grab a key message from this universe of data.

That's where data visualization comes in: summarizing and presenting large data in simple and easy-to-understand visualizations to give readers insightful information.

There are many advanced visualizations (e.g., networks, 3D-models and map overlays) used for specialized purposes such as 3D medical imaging, urban transportation simulation, and disaster relief monitoring. But regardless of the complexity of a visualization, its purpose is to help readers see a pattern or trend in the data being analyzed, rather than having them read tedious descriptions such as: "A's profit was more than B by 2.9% in 2000, and despite a profit growth of 25% in 2001, A's profit became less than B by 3.5% in 2001." A good visualization summarizes information and organizes in a way that enables the reader to focus on the points that are relevant to the key message being conveyed.

An analysis clearly explained with tables, graphs, charts and diagrams, keeping in mind that creating a good visualization is an iterative process.

Visualization Example

To demonstrate how each of the visualization tools could be used, imagine an executive for a company who wants to analyze the sales performance of his division. Table 6.1 show the important raw sales data for the current year, alphabetically sorted by Product names.

Product	Revenue	Orders	SalesPers
AA	9731	131	23
BB	355	43	8
CC	992	32	6
DD	125	31	4

EE	933	30	7
FF	676	35	6
CG	1411	128	13
HH	5116	132	38
JJ	215	7	2
KK	3833	122	50
LL	1348	15	7
MM	1201	28	13

Table 6.1: Raw Performance Data

To reveal some meaningful pattern, a good first step would be to sort the table by Product revenue, with highest revenue first. We could total up the values of Revenue, Orders, and Sales persons for all products. We can also add some important ratios to the right of the table (Table 6.2).

Product	Revenue	Orders	SalesPers	Rev/Order	Rev/Sales P	Orders/Sales P
AA	9731	131	23	74.3	423:1	5.7
HH	5116	132	38	38.8	134.6	3.5
KK	3333	122	50	31.4	76.7	2.4
GG	1411	128	13	11.0	108.5	9.8
LL	1348	15	7	89.9	192.6	2.1
MM	1201	28	13	42.9	92.4	2.2
CC	992	32	6	31.0	165.3	5.3
EE	933	30	7	31.1	133.3	4.3
FF	676	35	6	19.3	112.7	5.8
BB	388	43	8	8.3	44.4	5.4
JJ	215	7	2	30.7	107.5	3.5
DD	125	31	4	4.0	31.3	7.8
Total	25936	734	177	35.3	146.5	4.1

Table 6.2: Sorted data, with additional ratios

There are too many numbers on this table to visualize any trends in them. The numbers are in different scales so plotting them on the same chart would not be easy. E.g. the Revenue numbers are in thousands while the SalesPers numbers and Orders/SalesPers are in the single or double digit.

One could start by visualizing the revenue as a pie-chart. The revenue proportion drops significantly from the first product to the next. (Figure 6.1).

It is interesting to note that the top 3 products produce almost 75% of the revenue.

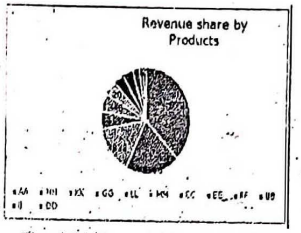


Figure 6.1: Revenue Share by Product

The number of orders for each product can be plotted as a bar graph . This shows that while the revenue is widely different for the top four products, they have approximately the same number of orders.

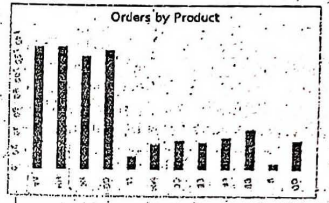


Figure 6.2: Orders by Products

Therefore, the orders data could be investigated further to see order patterns. Suppose additional data is made available for Orders by their size. Suppose the orders are chunked into 4 sizes: Tiny, Small, Medium, and Large. Additional data is shown in Table 6.3.

Product	Total Orders	Tiny	Small	Medium	Large
AA	131	5	44	70	12
HH	132	38	60	30	4
KK	122	20	50	44	8
GG	128	52	70	6	0
LL	15	2	3	5	5
MM	28	8	12	6	2
CC	32	5	17	10	0
EE	30	6	14	10	0
FF	35	10	22	3	0
BB	43	18	25	0	0
JJ	7	4	2	1	0
DD	31	21	10	0	0
Total	734	189	329	185	31

Table 6.3: Additional data on order sizes

Figure 6.3 is a stacked bar graph that shows the percentage of Orders by size for each product. This chart (Figure 6.3) brings a different set of insights. It shows that the product HH has

a larger proportion of tiny orders. The products at the far right have a large number of tiny orders and very few large orders.

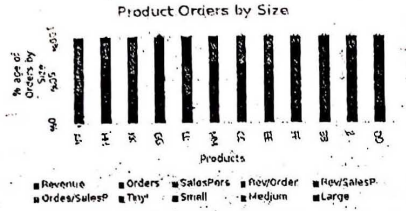


Figure 6.3: Product Orders by Order Size

Visualization Example phase -2

The executive wants to understand the productivity of salespersons. This analysis could be done both in terms of the number of orders, or revenue, per salesperson. There could be two separate graphs, one for the number of orders per salesperson, and the other for the revenue per salesperson. However, an interesting way is to plot both measures on the same graph to give a more complete picture. This can be done even when the two data have different scales. The data is here resorted by number of orders per salesperson.

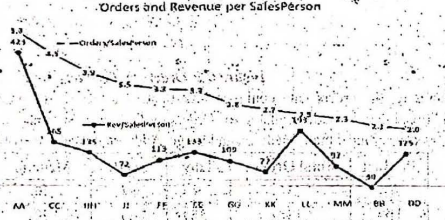


Figure 6.4: Salesperson productivity by product

Figure 6.4 shows two line graphs superimposed upon each other. One line shows the revenue per salesperson, while the other shows the number of orders per salesperson. It shows that the highest productivity of 5.3 orders per sales person, down to 2.1 orders per salesperson. The second line, the blue line shows the revenue per sales person for each for the products. The revenue per salesperson is highest at 630, while it is lowest at just 30. And thus additional layers of data visualization can go on for this data set.

c. **What are some key requirements for good visualization.** **(04 Marks)**

Ans. To help the client in understanding the situation, the following are some key requirements for good visualization :

1. ***Fetch appropriate and correct data for analysis.*** This requires some understanding of the domain of the client and what is important for the client. E.g. in a business setting, one may need to understand the many measure of profitability and productivity.
2. ***Sort the data in the most appropriate manner.*** It could be sorted by numerical variables, or alphabetically by name.

3. ***Choose appropriate method to present the data.*** The data could be presented as a table, or it could be presented as any of the graph types.
4. ***The data set could be pruned*** **to include only the more significant elements.** More data is not necessarily better, unless it makes the most significant impact on the situation.
5. ***The visualization could show additional dimension for reference*** such as the expectations or targets with which to compare the results.
6. ***The numerical data may need to be binned into a few categories.*** E.g. the orders per person were plotted as actual values, while the order sizes were binned into 4 categorical choices.
7. ***High-level visualization could be backed by more detailed analysis.*** For the most significant results, a drill-down may be required.
8. ***There may be need to present additional textual information*** to tell the whole story. For example, one may require notes to explain some extraordinary results.

Module -4

7. a. **What is pruning? What are pre-pruning and post-pruning? Why choose one over the other?** **(08 Marks)**

Ans. **Pruning :** The tree could be trimmed to make it more balanced and more easily usable. The pruning is often done after the tree is constructed, to balance out the tree and improve usability. The symptoms of an overfitted tree are a tree too deep, with too many branches, some of which may reflect anomalies due to noise or outliers. Thus, the tree should be pruned. There are two approaches to avoid over-fitting.

- Pre-pruning means to halt the tree construction early, when certain criteria are met. The downside is that it is difficult to decide what-criteria to use for halting the construction, because we do not know what may happen subsequently, if we keep growing the tree.
- Post-pruning: Remove branches or sub-trees from a "fully grown" tree. This method is commonly used. C4.5 algorithm uses a statistical method to estimate the errors at each node for pruning. A validation set may be used for pruning as well.

The most popular decision tree algorithms are C5, CART and CHAID (Table 7.1)

Table 7.1: Comparing popular Decision Tree algorithms

Decision Tree	C4.5	CART	CHAID
Full name	Iterative Dichotomiser (ID3)	Classification and regression trees	Chi-square automatic interaction detector
Basic algorithm	Hunt's algorithm	Hunt's algorithm	Adjusted significance testing
Developer	Ross Quinlan	Bremman	Gordon Kass
When developed	1986	1984	1980
Types of trees	Classification	Classification and regression trees	Classification and regression
Serial implementation	Tree growth and tree pruning	Tree growth and tree pruning	Tree growth and tree pruning
Type of data	Discrete and continuous; incomplete data	Discrete and continuous	Non-normal data also accepted

Types of splits	Multiway splits	Binary splits only; clever surrogate splits to reduce tree depth	Multiway splits as default
Splitting criteria	Information gain	Gini coefficient, and others	Chi-square test
Pruning criteria	Clever bottom up technique avoids overfitting	Remove weakest Links first	Trees can become very large
Implementation	Publicly available	Publicly available in most packages	Popular in market research, for segmentation

b. **Using the data that follows, create a regression model to predict a house price from the size of the house. Here are sample house data:** **(08 Marks)**

House Price	Size (sqft)
$229,500	1,850
$273,300	2,190
$247,000	2,100
$195,100	1,930
$261,000	2,300
$179,700	1,710
$168,500	1,550
$234,400	1,920
$168,500	1,840
$180,400	1,720
$156,200	1,660
$288,350	2,405
$156,750	1,525
$202,100	2,030
$256,800	2,240

Ans. The regression model is described as a linear equation that follows. y is the dependent variable, that is, the variable being predicted. x is the independent variable, or the predictor variable. There could be many predictor variables (such as $x1, x2, \ldots$) in a regression equation. However, there can be only one dependent variable (y) in the regression equation.

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Here are sample house data:

House Price	Size (sqft)
$229,500	1,850
$273,300	2,190
$247,000	2,100
$195,100	1,930
$261,000	2,300
$179,700	1,710
$168,500	1,550
$234,400	1,920

$168,500	1,840
$180,400	1,720
$156,200	1,660
$288,350	2,405
$156,750	1,525
$202,100	2,030
$256,800	2,240

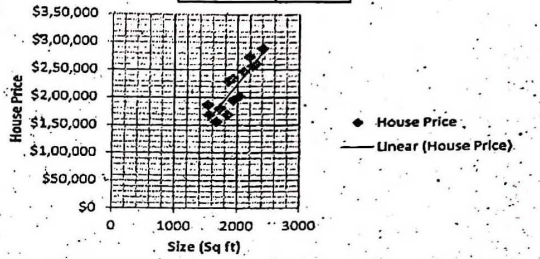


Figure 7.1 Scatter plot and regression equation between House price and house size

The two dimensions of (one predictor, one outcome variable) data can be plotted on a scatter diagram. A scatter plot with a best-fitting line looks like the graph that follows (Figure 7.1). Visually, one can see a positive correlation between house price and size (sqft). However, the relationship is not perfect. Running a regression model between the two variables produces the following output (truncated).

Regression Statistics	
Multiple r	0.891
r^2	0.794
	Coefficients
Intercept	-54,191
Size (sqft)	139.48

It shows the coefficient of correlation is 0.891. $r2$, the measure of total variance explained by the equation, is 0.794, or 79 percent. That means the two variables are moderately and positively correlated.

Regression coefficients help create the following equation for predicting house prices.

House Price ($) = 139.48 × Size (sqft) − 54,191

This equation explains only 79 percent of the variance in house prices.

Suppose other predictor variables are made available, such as the number of rooms in the house, it might help improve the regression model. The house data now looks like this:

House Price	Size (sqft)	# Rooms
$229,500	1,850	4
$273,300	2,190	5
$247,000	2,100	4

$195,100	1,930	3
$261,000	2,300	4
$179,700	1,710	2
$168,500	1,550	2
$234,400	1,920	4
$168,500	1,840	2
$180,400	1,720	2
$156,200	1,660	2
$288,350	2,405	5
$156,750	1,525	3
$202,100	2,030	2
$256,800	2,240	4

While it is possible to make a three-dimensional scatter plot, one can alternatively examine the correlation matrix among the variables.

	House Price	Size (sq ft)	# Rooms
House Price	1		
Size (sq ft)	0.891	1	
Rooms	0.944	0.748	1

It shows that the house price has a strong correlation with number of rooms (0.944) as well. Thus, it is likely that adding this variable to the regression model will add to the strength of the model. Running a regression model between these three variables produces the following output.

Regression Statistics	
Multiple r	0.984
r^2	0.968
	Coefficients
Intercept	12,923
Slie(sqft)	65.60
Rooms	23,613

It shows the coefficient of correlation of this regression model is 0.984. $r2$, the total variance explained by the equation, is 0.968, or 97 percent. That means the variables are positively and very strongly correlated.

Adding a new relevant variable has helped improve the strength of the regression model.

Using the regression coefficients helps create the following equation for predicting house prices.

House Price ($) = 65.6 × Size (sqft) + 23,613 × Rooms + 12,924

This equation shows a 97-percent goodness-of-fit with the data, which is very good for business and economic data. There is always some random variation in naturally occurring business data, and it is not desirable to overfit the model to the data.

This predictive equation should be used for future transactions. Given a situation that follows, it will be possible to calculate the price of the house with 2,000 sqft and 3 rooms.

House Price	Size (sq ft)	# Rooms
?	2,000	3

House Price ($) = 65.6 × 2,000 (sqft) + 23,613 × 3 + 12,924 = $214,963

The predicted values should be compared to the actual values to see how close the model is able to predict the actual value, As new data points become available, there are opportunities to fine-tune and improve the model.

OR

8. a. **What makes a neural network versatile enough for supervised as well as non-supervised learning tasks?** **(08 Marks)**

Ans. **Supervised Learning**

- Training data includes both the input and the desired results.
- For some examples the correct results (targets) are known and are given in input to the model during the learning process.
- The construction of a proper training, validation and test set (Bok) is crucial.
- These methods are usually fast and accurate.
- Have to be able to generalize: give the correct results when new data are given in input without knowing a priori the target.

Supervised learning is based on training a data sample from data source with correct classification already assigned. Such techniques are utilized in feedforward or MultiLayer Perceptron (MLP) models. These MLP has three distinctive characteristics:

1. One or more layers of hidden neurons that are not part of the input or output layers of the network that enable the network to learn and solve any complex problems
2. The nonlinearity reflected in the neuronal activity is differentiable and,
3. The interconnection model of the network exhibits a high degree of connectivity

These characteristics along with learning through training solve difficult and diverse problems. Learning through training in a supervised ANN model also called as error back-propagation algorithm. The error correction-learning algorithm trains the network based on the input-output samples and finds error signal, which is the difference of the output calculated and the desired output and adjusts the synaptic weights of the neurons that is proportional to the product of the error signal and the input instance of the synaptic weight. Based on this principle, error back propagation learning occurs in two passes:

Forward Pass: Here, input vector is presented to the network. This input signal propagates forward, neuron by neuron through the network and emerges at the output end of the network as output signal: $y(n) = \varphi(v(n))$ where $v(n)$ is the induced local field of a neuron defined by $v(n) = \Sigma\, w(n)y(n)$. The output that is calculated at the output layer $o(n)$ is compared with the desired response $d(n)$ and finds the error $e(n)$ for that neuron. The synaptic weights of the network during this pass are remains same.

Backward Pass: The error signal that is originated at the output neuron of that layer is propagated backward through network. This calculates the local gradient for each neuron in each layer and allows the synaptic weights of the network to undergo changes in accordance with the delta rule as:

$\Delta w(n) = \eta * \delta(n) * y(n)$.

This recursive computation is continued, with forward pass followed by the backward pass for each input pattern till the network is converged.

Supervised learning paradigm of an ANN is efficient and finds solutions to several linear and non-linear problems such as classification, plant control, forecasting, prediction, robotics etc

Unsupervised Learning

- The model is not provided with the correct results during the training.
- Can be used to cluster the input data in classes on the basis of their statistical properties

only.

- Cluster significance and labeling.
- The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

Self-Organizing neural networks learn using unsupervised learning algorithm to identify hidden patterns in unlabelled input data. This unsupervised refers to the ability to learn and organize information without providing an error signal to evaluate the potential solution. The lack of direction for the learning algorithm in unsupervised learning can sometime be advantageous, since it lets the algorithm to look back for patterns that have not been previously considered . The main characteristics of Self-Organizing Maps (SOM) are:

1. It transforms an incoming signal pattern of arbitrary dimension into one or 2 dimensional map and perform this transformation adaptively
2. The network represents feedforward structure with a single computational layer consisting of neurons arranged in rows and columns.
3. At each stage of representation, each input signal is kept in its proper context and,
4. Neurons dealing with closely related pieces of information are close together and they communicate through synaptic connections.

The computational layer is also called as competitive layer since the neurons in the layer compete with each other to become active. Hence, this learning algorithm is called competitive algorithm. Unsupervised algorithm in SOM works in three phases:

Competition phase: for each input pattern x, presented to the network, inner product with synaptic weight w is calculated and the neurons in the competitive layer finds a discriminant function that induce competition among the neurons and the synaptic weight vector that is close to the input vector in the Euclidean distance is announced as winner in the competition. That neuron is called best matching neuron, i.e. $x = \arg\min \|x - w\|$.

Cooperative phase: the winning neuron determines the center of a topological neighborhood h of cooperating neurons. This is performed by the lateral interaction d among the cooperative neurons. This topological neighborhood reduces its size over a time period.

Adaptive phase: enables the winning neuron and its neighborhood neurons to increase their individual values of the discriminant function in relation to the input pattern through suitable synaptic weight adjustments, $\Delta w = \eta h(x)(x - w)$.

Upon repeated presentation of the training patterns, the synaptic weight vectors tend to follow the distribution of the input patterns due to the neighborhood updating and thus ANN learns without supervisor [2].

Self-Organizing Model naturally represents the neuro-biological behavior, and hence is used in many real world applications such as clustering, speech recognition, texture segmentation, vector coding etc

b. Identify the clusters from the data as shown in Dataset below table.Determine the number of clusters and the center points of those clusters. (08 Marks)

X	Y
2	4
2	6
5	6
4	7
8	3
5	6
5	2

5	7
6	3
4	4

Ans. A scatter plot of 10 data points in 2 dimensions shows them distributed fairly randomly (Figure 8.1). As a bottom-up technique, the number of clusters and their centroids can be intuited.

The points are distributed randomly enough that it could be considered as one cluster. The circle would represent the central point (centroid) of these points. However, there is a big distance between the points (2,6) and (8,3). So, this data could be broken into two clusters. The three points at the bottom right could form one cluster and the other seven could form the other cluster. The two clusters would look like this (Figure 8.2). The circles will be the new centroids.

The bigger cluster seems too far apart. So, it seems like the four points on the top will form a separate cluster. The three clusters could look like this (Figure 8.3).

This solution has three clusters. The cluster on the right is far from the other two clusters. However, its centroid is not too close to all the data points. The cluster at the top looks very tight-fitting, with a nice centroid. The third cluster, at the left, is spread out and may not be of much usefulness.

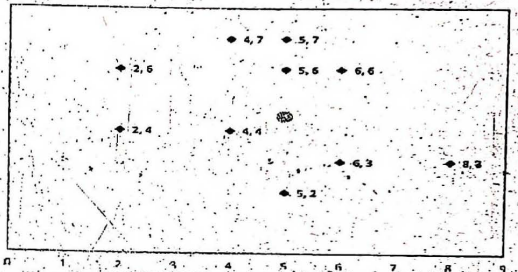


Figure 8.1 Initial data points and the centroid (shown as thick dot)

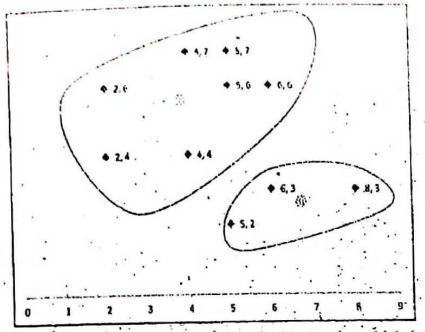


Figure 8.2Dividing into two clusters (centroids shown as thick dots)

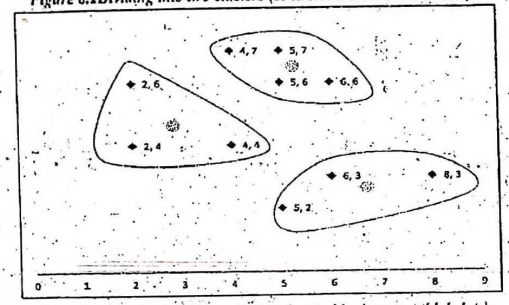


Figure 8.3 Dividing into three clusters (centroids shown as thick dots)

This was an exercise in producing three best-fitting cluster definitions from the given data. The right number of clusters will depend on the data and the application for which the data would be used.

K-Means Algorithm for Clustering

K-means is the most popular clustering algorithm. It iteratively computes the clusters and their centroids. It is a top-down approach to clustering. Starting with a given number of K clusters, say 3 clusters; thus, three random centroids will be created as starting points of the centers of three clusters (Figure 8.4). The circles are initial cluster centroids.

Step 1: For a data point, distance values will be from each of the three centroids. The data point will be assigned to the cluster with the shortest distance to the centroid. All data points will thus be assigned to one data point or the other. The arrows from each data element show the centroid that the point is assigned to (Figure 8.5).

Step 2: The centroid for each cluster will now be recalculated such that it is closest to all the data points allocated to that cluster. The dashed arrows show the centroids being moved from

their old (shaded) values to the revised new values (Figure 8.7).

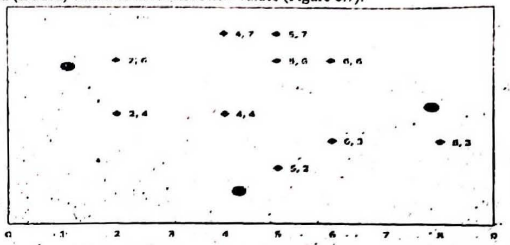


Figure 8.4 Randomly assigning three centroids for three data clusters

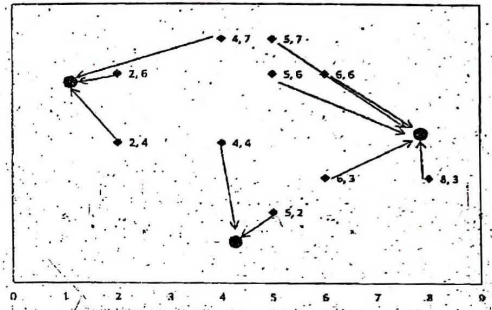


Figure 8.5 Assigning data points to closest centroid

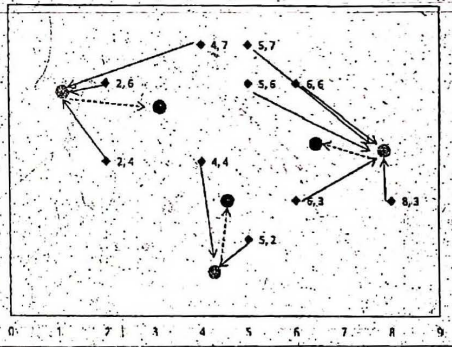


Figure 8.6 Recomputing centroids for each cluster

Step 3: Once again, data points are assigned to the three centroids closest to it (Figure 8.7). The new centroids will be computed from the data points in the cluster until finally the centroids stabilize in their locations. These are the three clusters computed by this algorithm (Figure 8 8).

The three clusters shown are a 3-datapoints cluster with centroid (6.5,4.5), a 2-datapoint cluster with centroid (4.5,3), and a 5-datapoint cluster with centroid (3.5,3).

These cluster definitions are different from the ones derived visually. This is a function of the random starting centroid values. The centroid points used earlier in the visual exercise were different from that chosen with the K-means clustering algorithm. The K-means clustering exercise should, therefore, be run again with this data, but with new random centroid starting values. With many runs, the cluster definitions are likely to stabilize. If the cluster definitions do not stabilize, that may be a sign that the number of clusters chosen is too high or too low. The algorithm should also be run with different values of K.

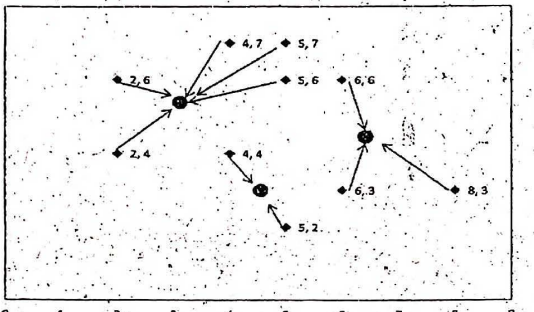


Figure 8.7 Assigning data points to Recomputed centroids

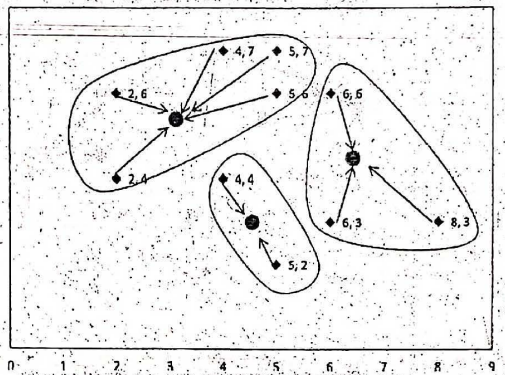


Figure 8.8 Recomputing centroids for each cluster till clusters stabilize

Here is the pseudocode for implementing a K-means algorithm. Algorithm K-Means (K number of clusters, D list of data points).

1. Choose K number of random data points as initial centroids (cluster centers).
2. Repeat till cluster centers stabilize:
   a. Allocate each point in D to the nearest of K centroids.
   b. Compute centroid for the cluster using all points in the cluster.

Selecting the Number of Clusters

The correct choice of the value of K is often ambiguous. It depends on the shape and scale of the distribution points in a data set and the desired clustering resolution of the user. Heuristics are needed to pick the right number. One can graph the percentage of variance explained by the clusters against the number of clusters. The first clusters will add more information, but at some point the marginal gain in variance will fall, giving a sharp angle to the graph, looking like an elbow.

At that elbow point, adding more clusters will not add much incremental value. That would be the desired value of K (Figure 8.9).

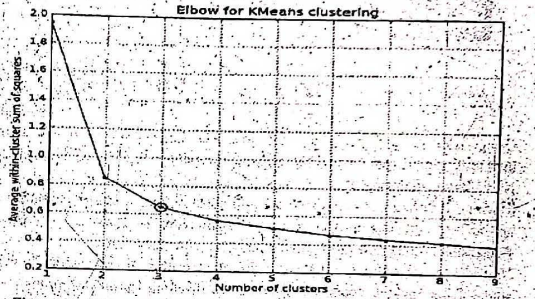


Figure 8.9 Elbow method for determining number of clusters in a data set

To engage with the data and to understand the clusters better, it is often better to start with a small number of clusters, such as 2 or 3, depending upon the data set and the application domain. The number can be increased subsequently, as needed from an application point of view.

This helps understand the data and the clusters progressively better.

Module-5

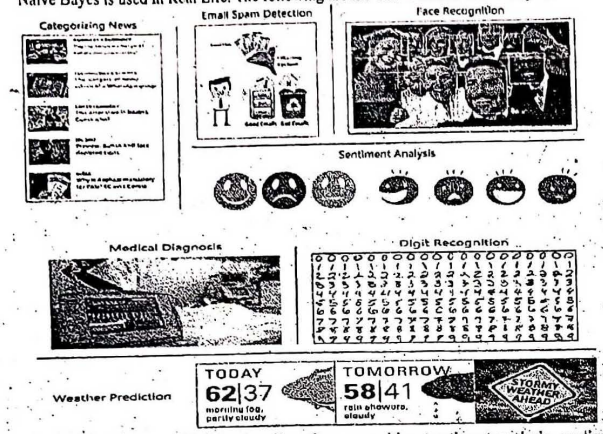
9. a. **What are the most popular applications of Naive-Bayes techniques?** **(04 Marks)**

Ans. Naive Bayes classifiers is a machine learning algorithm. If you wonder, how Google marks some of the mails as spam in your inbox, a machine learning algorithm will be used to classify an incoming email as spam or not spam.

Ans. Some of real world examples are as given below

- To mark an email as spam, or not spam ?
- Classify a news article about technology, politics, or sports ?
- Check a piece of text expressing positive emotions, or negativeemotions?
- Also used for face recognition softwares

Naive Bayes is used in Real Life. The following are the Use Cases of Naive Bayes:



Categorizing news, email spam detection, face recognition, sentiment analysis, medical diagnosis, digit recognition and weather prediction are just few of the popular use cases of Naive Bayes algorithm.

Machine Learning explores the study and construction of algorithms that can learn from and make predictions on data. Among Classification Algorithms, *Naive Bayes* along with Regression is one of the most popular and powerful algorithms.

Naive Bayes classifiers is a machine learning algorithm. If you wonder, how Google marks some of the mails as spam in your inbox, a machine learning algorithm will be used to classify an incoming email as spam or not spam.

b. What Is the Point in Using SVMs as a Classification Technique? (12 Marks)

Ans. All classification techniques have advantages and disadvantages, which are more or less important according to the data which are being analysed, and thus have a relative relevance. SVMs can be a useful tool for insolvency analysis, in the case of non-regularity in the data, for example when the data are not regularly distributed or have an unknown distribution. It can help evaluate information, i.e. financial ratios which should be transformed prior to entering the score of classical classification techniques.

The advantages of the SVM technique can be summarised as follows:

1. By introducing the kernel, SVMs gain flexibility in the choice of the form of the threshold separating solvent from insolvent companies, which needs not be linear and even needs not have the same functional form for all data, since its function is non-parametric and operates locally. As a consequence they can work with financial ratios, which show a non-monotone relation to the score and to the probability of default, or which are non-linearly dependent, and this without needing any specific work on each non-monotone variable.
2. Since the kernel implicitly contains a non-linear transformation, no assumptions about the functional form of the transformation, which makes data linearly separable, is necessary. The

transformation occurs implicitly on a robust theoretical basis and human expertise judgement beforehand is not needed.

3. SVMs provide a good out-of-sample generalization, if the parameters C and r (in the case of a Gaussian kernel) are appropriately chosen. This means that, by choosing an appropriate generalization grade, SVMs can be robust, even when the training sample has some bias.
4. SVMs deliver a unique solution, since the optimality problem is convex. This is an advantage compared to Neural Networks, which have multiple solutions associated with local minima and for this reason may not be robust over different samples.
5. With the choice of an appropriate kernel, such as the Gaussian kernel, one can put more stress on the similarity between companies, because the more similar the financial structure of two companies is, the higher is the value of the kernel. Thus when classifying a new company, the values of its financial ratios are compared with the ones of the support vectors of the training sample which are more similar to this new company. This company is then classified according to with which group it has the greatest similarity.

Here are some examples where the SVM can help coping with non-linearity and non-monotonicity. One case is, when the coefficients of some financial ratios in equation (1), estimated with a linear parametric model, show a sign that does not correspond to the expected one according to theoretical economic reasoning.

The reason for that may be that these financial ratios have a non-monotone relation to the PD and to the score. The unexpected sign of the coefficients depends on the fact, that data dominate or cover the part of the range, where the relation to the PD has the opposite sign. One of these financial ratios is typically the growth rate of a company, as pointed out by . Also leverage may show non-monotonicity, since if a company primary works with its own capital, it may not exploit all its external financing opportunities properly. Another example may be the size of a company: small companies are expected to be more financially instable; but if a company has grown too fast or if it has become too static because of its dimension, the big size may become a disadvantage. Because of these characteristics, the above mentioned financial ratios are often sorted out, when selecting the risk assessment model according to a linear classification technique. Alternatively an appropriate evaluation of this information in linear techniques requires a transformation of the input variables, in order to make them monotone and linearly separable. A common **disadvantage of non-parametric techniques** such as SVMs is the lack of transparency of results.

SVMs cannot represent the score of all companies as a simple parametric function of the financial ratios, since its dimension may be very high. It is neither a linear combination of single financial ratios nor has it another simple functional form. The weights of the financial ratios are not constant. Thus **the marginal contribution of each financial ratio to the score is variable.** Using a Gaussian kernel each company has its own weights according to the difference between the value of their own financial ratios and those of the support vectors of the training data sample.

Interpretation of results is however possible and can rely on graphical visualization, as well as on a local linear approximation of the score. The SVM threshold can be represented within a bi-dimensional graph for each pair of financial ratios. This visualization technique cuts and projects the multidimensional feature space as well as the multivariate threshold function separating solvent and insolvent companies on a bi-dimensional one, by fixing the values of the other financial ratios equal to the values of the company, which has to be classified. By this way, different companies will have different threshold projections.

However, an analysis of these graphs gives an important input about the direction towards

which the financial ratios of non-eligible companies should change, in order to reach eligibility.

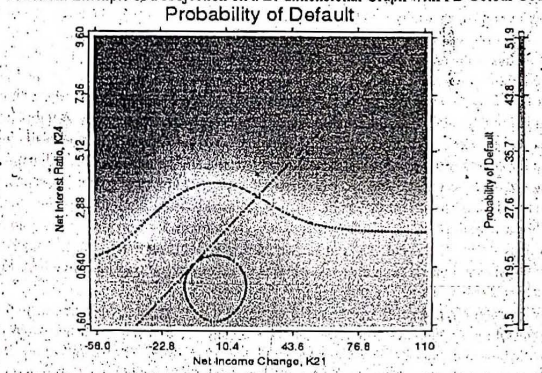
The PD can represent a third dimension of the graph, by means of isoquants and colour coding. The approach chosen for the estimation of the PD can be based on empirical estimates or on a theoretical model.

Since the relation between score and PD is monotone, a local linearization of the PD can be calculated for single companies by estimating the tangent curve to the isoquant of the score. For single companies this can offer interesting information about the factors influencing their financial solidity.

In the figure below the PD is estimated by means of a Gaussian kernel5 on data belonging to the trade sector and then smoothed and monotonized by means of a Pool Adjacent Violator algorithm.6 The pink curve represents the projection of the SVM threshold on a binary space with the two variables K21 (net income change) and K24 (net interest ratio), whereas all other variables are fixed at the level of company *j*. The blue curve represents the isoquant for the PD of company *j*, whose coordinates are marked by a triangle.

Figure . Graphical Visualization of the SVM Threshold and of a Local Linearization of the Score

Function: Example of a Projection on a Bi-dimensional Graph with PD Colour Coding



The grey line corresponds to the linear approximation of the score or PD function projection for company *j*. One interesting result of this graphical analysis is that successful companies with a low PD often lie in a closed space. This implies that there exists an optimal combination area for the financial ratios being considered, outside of which the PD gets higher. If we consider the net income change, we notice that its influence on the PD is non-monotone. Both too low or too high growth rates imply a higher PD. This may indicate the existence of the optimal growth rate and suggest that above a certain rate a company may get into trouble; especially if the cost structure of the company is not optimal i.e. the net interest ratio is too high. But if a company lies in the optimal growth zone, it can also afford a higher net interest ratio.

OR

10. a. **What are the two major ways that a website can become popular?** **(04 Marks)**

Ans. The Web works through a system of hyperlinks using the hypertext protocol (http). Any page can create a hyperlink to any other page, it can be linked to by another page. The intertwined or self-referral nature of web lends itself to some unique network analytical algorithms. The structure of Web pages could also be analyzed to examine the pattern of hyperlinks among pages. There are two basic strategic models for successful websites: Hubs and Authorities.

1. ***Hubs***: These are pages with a large number of interesting links. They serve as a hub, or a gathering point, where people visit to access a variety of information. Media sites like Yahoo.com, or government sites would serve that purpose. More focused sites like Traveladvisor.com and yelp.com could aspire to becoming hubs for new emerging areas.

2. ***Authorities***: Ultimately, people would gravitate towards pages that provide the most complete and authoritative information on a particular subject. This could be factual information, news, advice, user reviews etc. These websites would have the most number of inbound links from other websites. Thus Mayoclinic.com would serve as an authoritative page for expert medical opinion. NYtimes.com would serve as an authoritative page for daily news.

b. **Write short notes on web mining algorithm.** **(04 Marks)**

Ans. **Web Mining Algorithms**

Hyperlink-Induced Topic Search (HITS) is a link analysis algorithm that rates web pages as being hubs or authorities. Many other HITS-based algorithms have also been published. The most famous and powerful of these algorithms is the PageRank algorithm. Invented by Google co-founder Larry Page, this algorithm is used by Google to organize the results of its search function. This algorithm helps determine the relative importance of any particular web page by counting the number and quality of links to a page. The websites with more number of links, and/or more links from higher-quality websites, will be ranked higher. It works in a similar way as determining the status of a person in a society of people. Those with relations to more people and/or relations to people of higher status will be accorded a higher status. PageRank is the algorithm that helps determine the order of pages listed upon a Google Search query. The original PageRank algorithm formuation has been updated in many ways and the latest algorithm is kept a secret so other websites cannot take advantage of the algorithm and/manipulate their website according to it. However, there are many standard elements that remain unchanged. These elements lead to the principles for a good website. This process is also called Search Engine Optimization (SEO).

c. **Explain the Practical consideration of Social network analysis, Give the difference between Social Network Analysis v/s Traditional Data Analytics**

Ans. **PRATICAL CONSIDERATION:** **(08 Marks)**

Network Size :Most SNA research is done using small networks. Collecting data about large network can be very challenging. This is because the number of link is the order of the square of the number of nodes. Thus, in a network of 1000 nodes there are potentially 1 million possible pairs of links.

Gathering Data: Electronics communication records (email, chat, etc.) can be harnessed to gather social network data more easily. Data on the nature and quality of relationship need to be collected using survey documents. Capturing and cleansing and organizing the data can take a lot of time and effort , just like in a typical analytics project.

Computation And Visualization: Modeling large networks can be computationally challenging and visualizing them also would require special skills . Big data analytical tools may be needed to compute large networks.

Dynamic Networks: Relationships between nodes in a social network can be fluid. They can change in strength and functional nat6ure . For Example, there could be multiple relationships between two people... they could simultaneously be coworker , coauthors, and spouses, The network should be modeled frequently to see the dynamics of the network.

Table 10.1 Social Network Analysis w/s Traditional Data Analytics

Dimension	Social Network Analysis	Traditional Data Mining
Nature of learning	Unsupervised-learning	Supervised and Unsupervised learning
Analysis of goals	Hub nodes ,important nodes, and sub-networks	Key decision rule, cluster cent roids
Dataset structures	A graph of nodes and (directed) links	Rectangular data of variables and instances
Analysis techniques	Visualization with statistics; iterative graphical computation	Machine learning , statistics
Quality measurement	Usefulness is key criterion	Predictive accuracy for classification techniques

Eight Semester B.E. Degree Examination, CBCS - June / July 2019

Big Data Analytics

Time: 3 hrs. **Max. Marks: 80**

Note : *Answer any FIVE full questions, selecting ONE full question from each module.*

Module - 1

1. a. **How does the Hadoop MapReduce Data flow work or a word count program? Give an example.** **(08 Marks)**

Ans. In Hadoop, MapReduce is a computation that decomposes large manipulation jobs into individual tasks that can be executed in parallel across a cluster of servers. The results of tasks can be joined together to compute final results.

MapReduce consists of 2 steps:

Map Function – It takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (Key-Value pair).

Example – (Map function in Word Count)

Input	Set of data	Bus, Car, bus, car, train, car, bus, car, train, bus, TRAIN,BUS, buS, caR, CAR, car, BUS, TRAIN.
Output	Convert into another set of data (Key,Value)	(Bus,1), (Car,1), (bus,1), (car,1), (train,1), (car,1), (bus,1), (car,1), (train,1), (bus,1), (TRAIN,1),(BUS,1), (buS,1), (caR,1), (CAR,1), (car,1), (BUS,1), (TRAIN,1)

Reduce Function – Takes the output from Map as an input and combines those data tuples into a smaller set of tuples.

Example – (Reduce function in Word Count)

Input (output of Map function)	Set of Tuples	(Bus,1), (Car,1), (bus,1), (car,1), (train,1), (car,1), (bus,1), (car,1), (train,1), (bus,1), (TRAIN,1),(BUS,1), (buS,1), (caR,1), (CAR,1), (car,1), (BUS,1), (TRAIN,1)
Output	Converts into smaller set of tuples	(BUS,7), (CAR,7), (TRAIN,4)

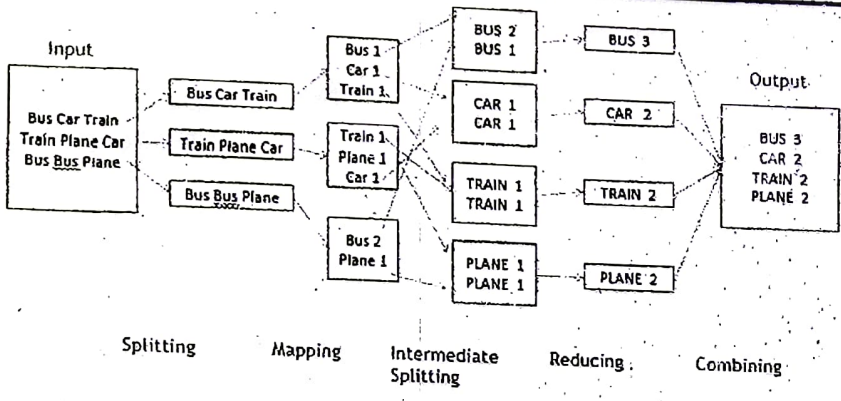


Fig. WorkFlow of MapReducing

Workflow of MapReduce consists of 5 steps:

1. Splitting – The splitting parameter can be anything, e.g. splitting by space, comma, semicolon, or even by a new line ('\n').

2. Mapping – as explained above.

3. Intermediate splitting – the entire process in parallel on different clusters. In order to group them in "Reduce Phase" the similar KEY data should be on the same cluster.

4. Reduce – it is nothing but mostly group by phase.

5. Combining – The last phase where all the data (individual result set from each cluster) is combined together to form a result.

```
package PackageDemo;
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
public class WordCount {
public static void main(String [] args) throws Exception
{
Configuration c=new Configuration();
String[] files=new GenericOptionsParser(c,args).getRemainingArgs();
Path input=new Path(files[0]);
Path output=new Path(files[1]);
Job j=new Job(c,"wordcount");
```

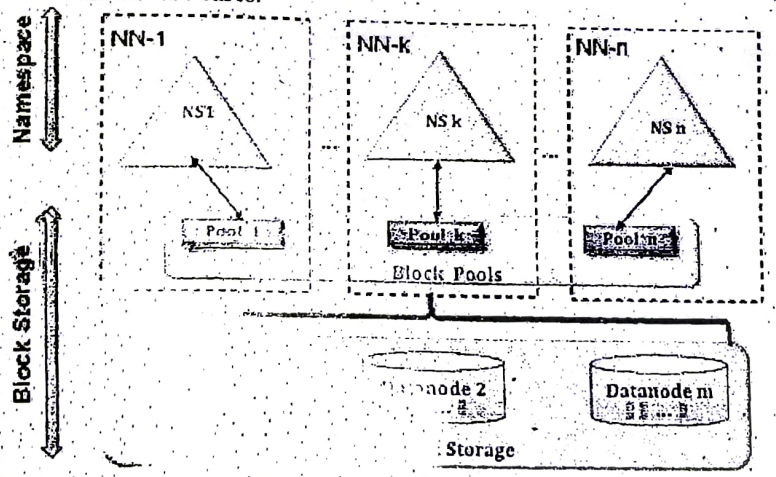


```
j.setJarByClass(WordCount.class);
j.setMapperClass(MapForWordCount.class);
j.setReducerClass(ReduceForWordCount.class);
j.setOutputKeyClass(Text.class);
j.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(j, input);
FileOutputFormat.setOutputPath(j, output);
System.exit(j.waitForCompletion(true)?0:1);
}
public static class MapForWordCount extends Mapper<LongWritable, Text, Text,
IntWritable>{
public void map(LongWritable key, Text value, Context con) throws IOException,
InterruptedException
{
String line = value.toString();
String[] words=line.split(",");
for(String word: words )
{
    Text outputKey = new Text(word.toUpperCase().trim());
 IntWritable outputValue = new IntWritable(1);
 con.write(outputKey, outputValue);
}
}
}
public static class ReduceForWordCount extends Reducer<Text, IntWritable, Text,
IntWritable>
{
public void reduce(Text word, Iterable<IntWritable> values, Context con) throws
IOException, InterruptedException
{
int sum = 0;
  for(IntWritable value : values)
  {
  sum += value.get();
  }
  con.write(word, new IntWritable(sum));
}
}
}
```

b. Briefly explain HDFS Name Node federation, NFS Gateway, snapshots, Checkpoint and Backups. (08 Marks)

Ans. **HDFS Federation** improves the existing HDFS architecture through a clear separation of namespace and storage, enabling generic block storage layer. It enables support for multiple namespaces in the cluster to improve scalability and isolation. Federation

also opens up the architecture, expanding the applicability of HDFS cluster to new implementations and use cases.



HDFS has two main

1. Namespace
   - Consists of directories, files and blocks.
   - It supports all the namespace related file system operations such as create, delete, modify and list files and directories.
2. Block Storage Service, which has two parts:
   - Block Management (performed in the Namenode)
     a. Provides Datanode cluster membership by handling registrations, and periodic heart beats.
     b. Processes block reports and maintains location of blocks.
     c. Supports block related operations such as create, delete, modify and get block location.
     d. Manages replica placement, block replication for under replicated blocks, and deletes blocks that are over replicated.
   - Storage - is provided by Datanodes by storing blocks on the local file system and allowing read/write access.

The prior HDFS architecture allows only a single namespace for the entire cluster. In that configuration, a single Namenode manages the namespace. HDFS Federation addresses this limitation by adding support for multiple Namenodes/namespaces to HDFS.

The NFS Gateway for HDFS allows clients to mount HDFS and interact with it through NFS, as if it were part of their local file system. The Gateway supports NFSv3. After mounting HDFS, a client user can perform the following tasks:

Browse the HDFS file system through their local file system on NFSv3 client-compatible operating systems.

Upload and download files between the HDFS file system and their local file system.

Stream data directly to HDFS through the mount point. File append is supported, but random write is not supported.

HDFS snapshots are similar to backups, but are created by administrator using the hdfs dfs-snapshot command. HDFS snapshots are read-only point-in-time copies of the file system. They offer the following features.

- Snapshots can be taken of a sub-tree of the file system or the entire file system.
- Snapshots can be used for data backups, protection against user errors, and disaster recovery.
- Snapshots creation is instantaneous.
- Blocks on the DataNodes are not copied, because the snapshot files record the block list and the file size. There is no data copying, although it appears to the user that there are duplicate files.
- Snapshots do not adversely affect regular HDFS operations.

A Checkpoint Node was introduced to solve the drawbacks of the NameNode. The changes are just written to edits and not merged to fsimage during the runtime. If the NameNode runs for a while edits gets huge and the next startup will take even longer because more changes have to be applied to the state to determine the last state of the metadata. The Checkpoint Node fetches periodically fsimage and edits from the NameNode and merges them. The resulting state is called checkpoint. After this is uploads the result to the NameNode.

There was also a similar type of node called "Secondary Node" but it doesn't have the "upload to NameNode" feature. So the NameNode need to fetch the state from the Secondary NameNode. It also was confussing because the name suggests that the Secondary NameNode takes the request if the NameNode fails which isn't the case.

Backup Node The Backup Node provides the same functionality as the Checkpoint Node, but is synchronized with the NameNode. It doesn't need to fetch the changes periodically because it receives a strem of file system edits. from the NameNode. It holds the current state in-memory and just need to save this to an image file to create a new checkpoint:

OR

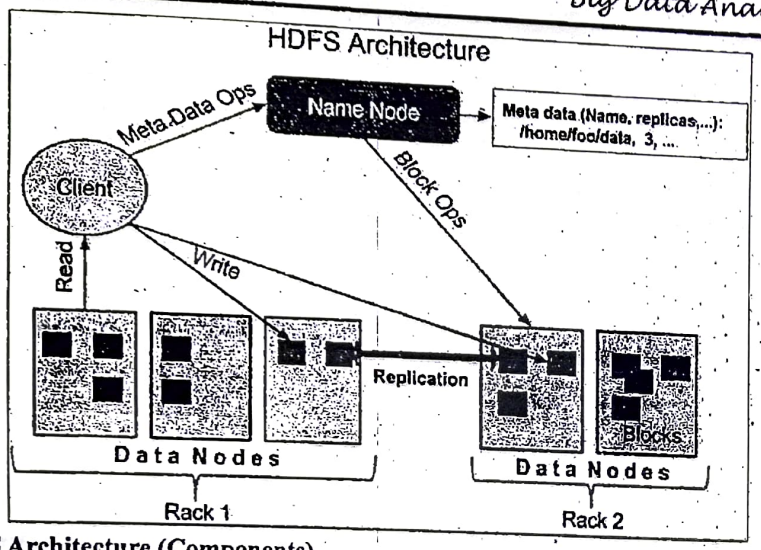
2. a. What do you understand by HDFS? Explain its components with a neat diagram. (10 Marks)

Ans. Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly faulttolerant and designed using low-cost hardware.

HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

Features of HDFS

a. It is suitable for the distributed storage and processing.
b. Hadoop provides a command interface to interact with HDFS.
c. The built-in servers of namenode and datanode help users to easily check the status of cluster.
d. Streaming access to file system data.
e. HDFS provides file permissions and authentication.



HDFS Architecture (Components)

Given above is the architecture of a Hadoop File System. HDFS follows the master-slave architecture and it has the following elements.

Namenode The namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software. It is a software that can be run on commodity hardware. The system having the namenode acts as the master server and it does the following tasks –

- Manages the file system namespace.
- Regulates client's access to files.
- It also executes file system operations such as renaming, closing, and opening files and directories.

Datanode The datanode is a commodity hardware having the GNU/Linux operating system and datanode software. For every node (Commodity hardware/System) in a cluster, there will be a datanode. These nodes manage the data storage of their system.

- Datanodes perform read-write operations on the file systems, as per client request.
- They also perform operations such as block creation, deletion, and replication according to the instructions of the namenode.

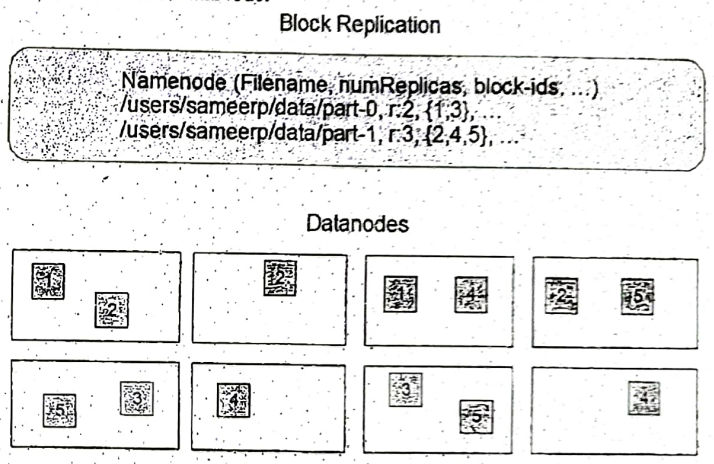
Block Generally the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

. **Bring out the concepts of HDFS block replication, with an example. (06 Marks)**

Data Replication HDFS is designed to reliably store very large files across machines in a large cluster. It stores each file as a sequence of blocks; all blocks in a file except the last block are the same size. The blocks of a file are replicated for fault tolerance. The block size and replication factor are configurable per file. An application can specify the

number of replicas of a file. The replication factor can be specified at file creation time and can be changed later. Files in HDFS are write-once and have strictly one writer at any time.

The NameNode makes all decisions regarding replication of blocks. It periodically receives a Heartbeat and a Blockreport from each of the DataNodes in the cluster. Receipt of a Heartbeat implies that the DataNode is functioning properly. A Blockreport contains a list of all blocks on a DataNode.



Module - 2

3. a. **Explain Apache Sqoop Import and Export method with neat diagrams. (10 Marks)**

Ans. Figure 1 describes the Sqoop data import (to HDFS) process. The data import is done in two steps. In the first step, shown in the figure, Sqoop examines the database to gather the necessary metadata for the data to be imported. The second step is a map-only (no reduce step) Hadoop job that Sqoop submits to the cluster. This job does the actual data transfer using the metadata captured in the previous step. Note that each node doing the import must have access to the database.

The imported data are saved in an HDFS directory. Sqoop will use the database name for the directory, or the user can specify any alternative directory where the files should be populated. By default, these files contain comma-delimited fields, with new lines separating different records. You can easily override the format in which data are copied over by explicitly specifying the field separator and record terminator characters. Once placed in HDFS, the data are ready for processing.

Data export from the cluster works in a similar fashion. The export is done in two steps, as shown in Figure 2. As in the import process, the first step is to examine the database for metadata. The export step again uses a map-only Hadoop job to write the data to the database. Sqoop divides the input data set into splits, then uses individual map tasks to push the splits to the database. Again, this process assumes the map tasks have access to the database.

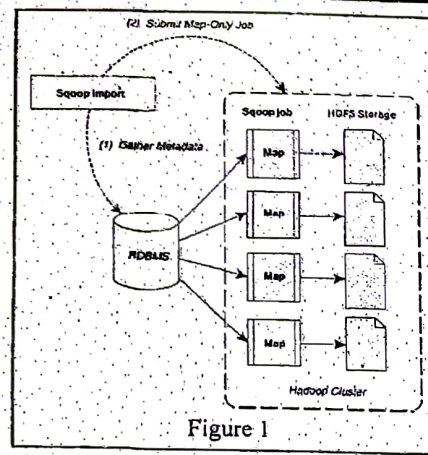


Figure 1

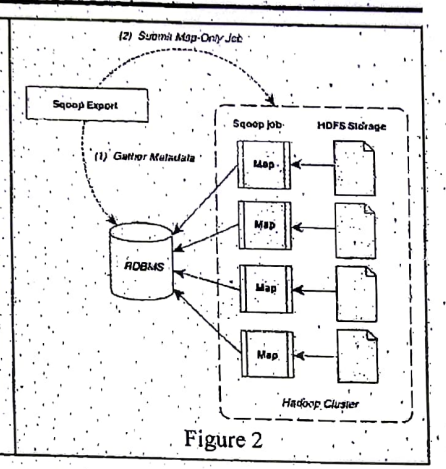


Figure 2

b. Explain with a neat diagram, the Apache Oozie work flow for Hadoop architecture. (06 Marks)

Ans. Oozie is a workflow director system designed to run and manage multiple related Apache Hadoop jobs. For instance, complete data input and analysis may require several discrete Hadoop jobs to be run as a workflow in which the output of one job serves as the input for a successive job. Oozie is designed to construct and manage these workflows. Oozie is not a substitute for the YARN scheduler. That is, YARN manages resources for individual Hadoop jobs, and Oozie provides a way to connect and control Hadoop jobs on the cluster.

Oozie workflow jobs are represented as directed acyclic graphs (DAGs) of actions. (DAGs are basically graphs that cannot have directed loops.) Three types of Oozie jobs are permitted:

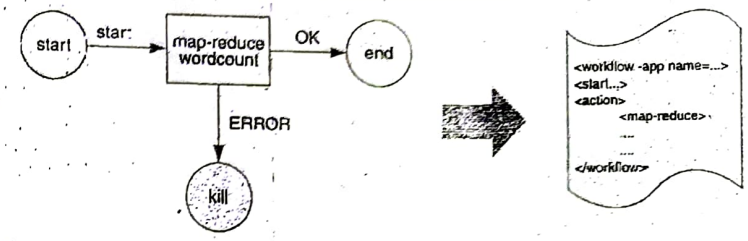
Image Workflow—a specified sequence of Hadoop jobs with outcome-based decision points and control dependency. Progress from one action to another cannot happen until the first action is complete.

Image Coordinator—a scheduled workflow job that can run at various time intervals or when data become available.

Image Bundle—a higher-level Oozie abstraction that will batch a set of coordinator jobs.

Oozie is integrated with the rest of the Hadoop stack, supporting several types of Hadoop jobs out of the box (e.g., Java MapReduce, Streaming MapReduce, Pig, Hive, and Sqoop) as well as system-specific jobs (e.g., Java programs and shell scripts). Oozie also provides a CLI and a web UI for monitoring jobs.

Below Figure depicts a simple Oozie workflow. In this case, Oozie runs a basic MapReduce operation. If the application was successful, the job ends; if an error occurred, the job is killed.

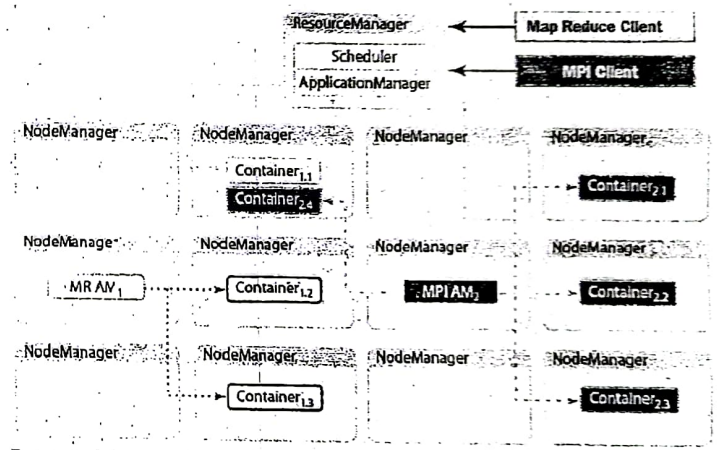


MapReduce Workflow DAG Workflow.xml

OR

4. a. How do you run Map Reduce and Message Passing Interface (MPI) on YARN architecture? Discuss. (10 Marks)

Ans.



The ResourceManager has a central and global view of all cluster resources and, therefore, can ensure fairness, capacity, and locality are shared across all users. Depending on the application demand, scheduling priorities, and resource availability, the ResourceManager dynamically allocates resource containers to applications to run on particular nodes. A container is a logical bundle of resources (e.g., memory, cores) bound to a particular cluster node. To enforce and track such assignments, the ResourceManager interacts with a special system daemon running on each node called the NodeManager. Communications between the ResourceManager and NodeManagers are heartbeat based for scalability. NodeManagers are responsible for local monitoring of resource availability, fault reporting, and container life-cycle management (e.g., starting and killing jobs). The ResourceManager depends on the NodeManagers for its "global view" of the cluster.

User applications are submitted to the ResourceManager via a public protocol and go through an admission control phase during which security credentials are validated and various operational and administrative checks are performed. Those applications

that are accepted pass to the scheduler and are allowed to run. Once the scheduler has enough resources to satisfy the request, the application is moved from an accepted state to a running state. Aside from internal bookkeeping, this process involves allocating a container for the single ApplicationMaster and spawning it on a node in the cluster. Often called container 0, the ApplicationMaster does not have any additional resources at this point, but rather must request additional resources from the ResourceManager.

The ApplicationMaster is the "master" user job that manages all application life-cycle aspects, including dynamically increasing and decreasing resource consumption (i.e., containers), managing the flow of execution (e.g., in case of MapReduce jobs, running reducers against the output of maps), handling faults and computation skew, and performing other local optimizations.

Above Figure illustrates the relationship between the application and YARN components. The YARN components appear as the large outer boxes (ResourceManager and NodeManagers), and the two applications appear as smaller boxes (containers), one dark and one light. Each application uses a different ApplicationMaster; the darker client is running a Message Passing Interface (MPI) application and the lighter client is running a traditional MapReduce application.

b. What do you understand by YARN Distributed-Shell? **(06 Marks)**

Ans. Distributed-Shell is an example application included with the Hadoop core components that demonstrates how to write applications on top of YARN. It provides a simple method for running shell commands and scripts in containers in parallel on a Hadoop YARN cluster.

The distributed shell client allows an application master to be launched that in turn would run the provided shell command on a set of containers.

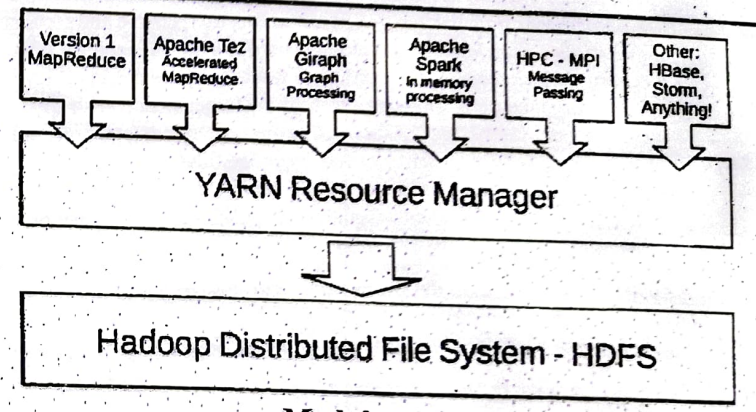
This client is meant to act as an example on how to write yarn-based applications.

To submit an application, a client first needs to connect to the ResourceManager aka ApplicationsManager or ASM via the ApplicationClientProtocol. The ApplicationClientProtocol provides a way for the client to get access to cluster information and to request for a new ApplicationId.

For the actual job submission, the client first has to create an ApplicationSubmissionContext. The ApplicationSubmissionContext defines the application details such as ApplicationId and application name, the priority assigned to the application and the queue to which this application needs to be assigned. In addition to this, the ApplicationSubmissionContext also defines the ContainerLaunchContext which describes the Container with which the ApplicationMaster is launched.

The ContainerLaunchContext in this scenario defines the resources to be allocated for the ApplicationMaster's container, the local resources (jars, configuration files) to be made available and the environment to be set for the ApplicationMaster and the commands to be executed to run the ApplicationMaster.

Using the ApplicationSubmissionContext, the client submits the application to the ResourceManager and then monitors the application by requesting the ResourceManager for an ApplicationReport at regular time intervals. In case of the application taking too long, the client kills the application by submitting a KillApplicationRequest to the ResourceManager.



Module - 3

5. a. Write any four Business Intelligence Application for various sectors. (08 Marks)

Ans. 1. Customer Relationship Management A business exists to serve a customer. A happy customer becomes a repeat customer. A business should understand the needs and sentiments of the customer, sell more of its offerings to the existing customers, and also, expand the pool of customers it serves. BI applications can impact many aspects of marketing.

a. Maximize the return on marketing campaigns: Understanding the customer's pain points from data-based analysis can ensure that the marketing messages are fine-tuned to better resonate with customers.

b. Improve customer retention (churn analysis): It is more difficult and expensive to win new customers than it is to retain existing customers. Scoring each customer on their likelihood to quit can help the business design effective interventions, such as discounts or free services, to retain profitable customers in a cost-effective manner.

c. Maximize customer value (cross-selling, upselling): Every contact with the customer should be seen as an opportunity to gauge their current needs. Offering a customer new products and solutions based on those imputed needs can help increase revenue per customer. Even a customer complaint can be seen as an opportunity to wow the customer. Using the knowledge of the customer's history and value, the business can choose to sell a premium service to the customer.

d. Identify and delight highly valued customers: By segmenting the customers, the best customers can be identified. They can be proactively contacted, and delighted, with greater attention and better service. Loyalty programs can be managed more effectively.

e. Manage brand image: A business can create a listening post to listen to social media chatter about itself. It can then do sentiment analysis of the text to understand the nature of comments and respond appropriately to the prospects and customers.

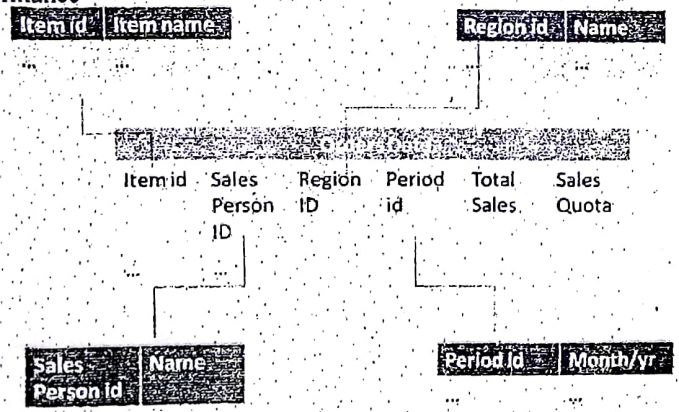
2. Health Care and Wellness Health care is one of the biggest sectors in advanced economies. Evidence- based medicine is the newest trend in data-based health

care management. BI applications can help apply the most effective diagnoses and prescriptions for various ailments. They can also help manage public health issues, and reduce waste and fraud.

3. **Education** As higher education becomes more expensive and competitive, it is a great user of data-based decision-making. There is a strong need for efficiency, increasing revenue, and improving the quality of student experience at all levels of education.
4. **Retail** organizations grow by meeting customer needs with quality products, in a convenient, timely, and cost-effective manner. Understanding emerging customer shopping patterns can help retailers organize their products, inventory, store layout, and web presence in order to delight their customers, which in turn would help increase revenue and profits. Retailers generate a lot of transaction and logistics data that can be used to solve problems.

b. Explain the star schema design of Data Warehousing with an example. **(06 Marks)**

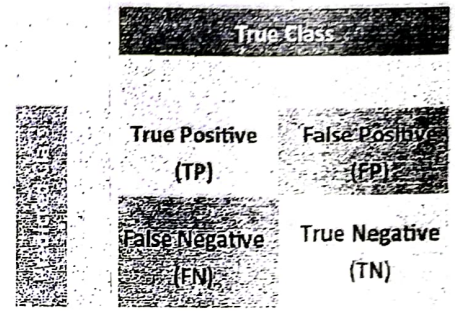
Ans. Star schema is the preferred data architecture for most DWs. There is a central fact table that provides most of the information of interest. There are lookup tables that provide detailed values for codes used in the central table. For example, the central table may use digits to represent a sales person. The lookup table will help provide the name for that sales person code. Here is an example of a star schema for a data mart for monitoring sales performance



Other schemas include the snowflake architecture. The difference between a star and snowflake is that in the latter, the lookup tables can have their own further lookup tables. There are many technology choices for ceveloping DW. This includes selecting the right database management system and the right set of data management tools. There are a few big and reliable providers of DW systems. The provider of the operational DBMS may be chosen for DW also. Alternatively, a best-of-breed DW vendor could be used. There are also a variety of tools out there for data migration, data upload, data retrieval, and data analysis.

c. What is Confusion Matrix. **(02 Marks)**

Ans.

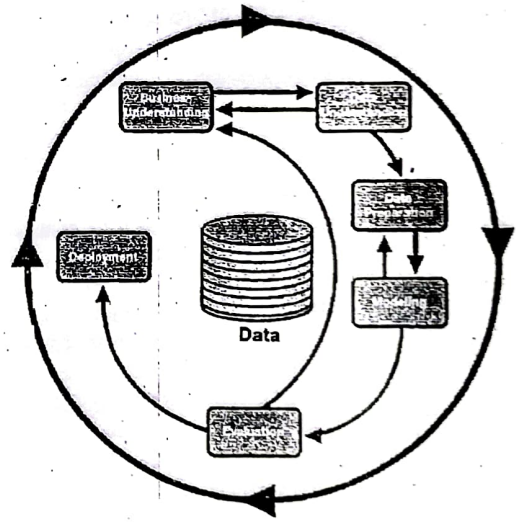


A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

OR

6. a. Explain CRISP-DM cycle with a neat diagram. **(08 Marks)**

Ans.



The data mining industry has proposed a Cross-Industry Standard Process for Data Mining (CRISP-DM). It has six essential steps

1. The first and most important step in data mining is business understanding, that is, asking the right business questions. A question is a good one if answering it would lead to large payoffs for the organization, financially and otherwise. In other words, selecting a data mining project is like any other project, in which it should show

strong payoffs if the project is successful. There should be strong executive support for the data mining project, which means that the project aligns well with the business strategy.

2. A second important step is to be creative and open in proposing imaginative hypotheses for the solution. Thinking outside the box is important, both in terms of a proposed model as well in the data sets available and required.
3. The data should be clean and of high quality. It is important to assemble a team that has a mix of technical and business skills, who understand the domain and the data. Data cleaning can take 60 to 70 percent of the time in a data mining project. It may be desirable to add new data elements from external sources of data that could help improve predictive accuracy.
4. Patience is required in continuously engaging with the data until the data yields some good insights. A host of modeling tools and algorithms should be used. A tool could be tried with different options, such as running different decision tree algorithms.
5. One should not accept what the data says at first. It is better to triangulate the analysis by applying multiple data mining techniques and conducting many what-if scenarios, to build confidence in the solution. Evaluate the model's predictive accuracy with more test data.
6. The dissemination and rollout of the solution is the key to project success. Otherwise the project will be a waste of time and will be a setback for establishing and supporting a data-based decision-process culture in the organization. The model should be embedded in the organization's business processes.

b. What do you understand by the term Data Visualization? How it is important in Big Data Analytics? **(05 Marks)**

Ans. As data and insights grow in number, a new requirement is the ability of the executives and decision makers to absorb this information in real time. There is a limit to human comprehension and visualization capacity. That is a good reason to prioritize and manage with fewer but key variables that relate directly to the key result areas of a role.

Here are few considerations when presenting data:

1. Present the conclusions and not just report the data.
2. Choose wisely from a palette of graphs to suit the data.
3. Organize the results to make the central point stand out.
4. Ensure that the visuals accurately reflect the numbers. Inappropriate visuals can create misinterpretations and misunderstandings.
5. Make the presentation unique, imaginative, and memorable.

c. Differentiate between Data Mining and Data Warehousing. **(03 Marks)**

Ans.

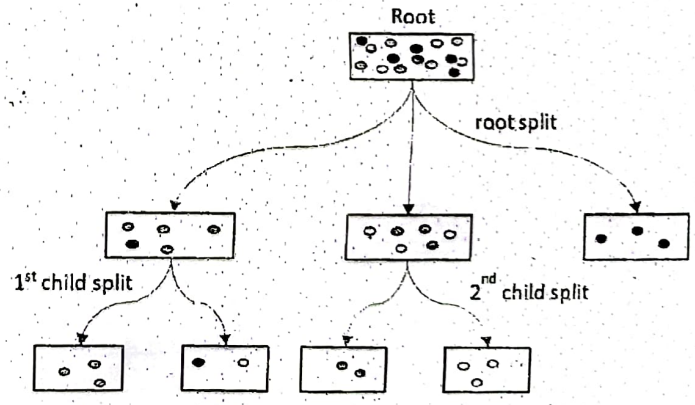
Data Mining	Data Warehouse
Data mining is the process of analyzing unknown patterns of data.	A data warehouse is database system which is designed for analytical instead of transactional work.

Data mining is a method of comparing large amounts of data to finding right patterns.	Data warehousing is a method of centralizing data from different sources into one common repository.
Data mining is usually done by business users with the assistance of engineers.	Data warehousing is a process which needs to occur before any data mining can take place.
Data mining is the considered as a process of extracting data from large data sets.	On the other hand, Data warehousing is the process of pooling all relevant data together.
One of the most important benefits of data mining techniques is the detection and identification of errors in the system.	One of the pros of Data Warehouse is its ability to update consistently. That's why it is ideal for the business owner who wants the best and latest features.

Module - 4

7. a. What is splitting variable? Describe three criteria for choosing a splitting variable. **(04 Marks)**

Ans. Decision trees are trained by passing data down from a root node to leaves. The data is repeatedly split according to predictor variables so that child nodes are more "pure" (i.e., homogeneous) in terms of the outcome variable. This process is illustrated below:



Three criteria for choosing a splitting variable.

a) Which variable to use for the first split? How should one determine the most important variable for the first branch, and subsequently, for each subtree? There are many measures like least errors, information gain, and Gini coefficient.
b) What values to use for the split? If the variables have continuous values, such as for age or BP, what value-ranges should be used to make bins?
c) How many branches should be allowed for each node? There could be binary trees, with just two branches at each node. Or there could be more branches allowed.

b. List some of the advantages and disadvantages of Regression Model. (04 Marks)

Ans. Advantage:

1. Regression models are easy to understand as they are built upon basic statistical principles, such as correlation and least square error.
2. Regression models provide simple algebraic equations that are easy to understand and use.
3. The strength (or the goodness of fit) of the regression model is measured in terms of the correlation coefficients, and other related statistical parameters that are well understood.

Disadvantage:

1. Regression models cannot cover for poor data quality issues. If the data is not prepared well to remove missing values, or is not well-behaved in terms of a normal distribution, the validity of the model suffers.
2. Regression models suffer from collinear problems (meaning strong linear correlations among some independent variables). If the independent variables have strong correlations among themselves, then they will eat into each other's predictive power and the regression coefficients will lose their ruggedness.
3. Regression models will not automatically choose between highly collinear variables, although some packages attempt to do that. Regression models can be unwieldy and unreliable if a large number of variables are included in the model. All variables entered into the model will be reflected in the regression equation, irrespective of their contribution to the predictive power of the model. There is no concept of automatic pruning the model.

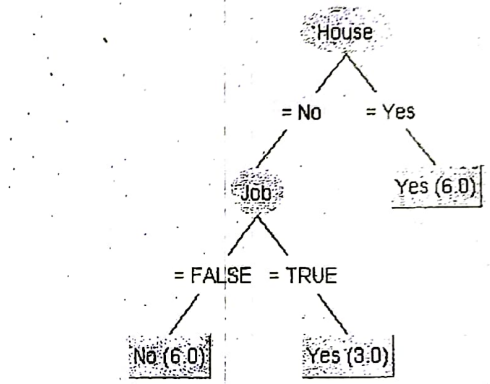
c. Create a decision tree for the following data set.

Age	Job	House	Credit	Loan Approved
Young	FALSE	No	Fair	No
Young	FALSE	No	Good	No
Young	TRUE	No	Good	Yes
Young	TRUE	Yes	Fair	Yes
Young	FALSE	No	Fair	No
Middle	FALSE	No	Fair	No
Middle	FALSE	No	Good	No
Middle	TRUE	Yes	Good	Yes
Middle	FALSE	Yes	Excellent	Yes
Middle	FALSE	Yes	Excellent	Yes
Old	FALSE	Yes	Excellent	Yes
Old	FALSE	Yes	Good	Yes
Old	TRUE	No	Good	Yes
Old	TRUE	No	Excellent	Yes
Old	FALSE	No	Fair	No

Then solve the following problem using the model.

Age	Job	House	Credit	Loan Approved
Young	FALSE	No	Good	???

Ans.



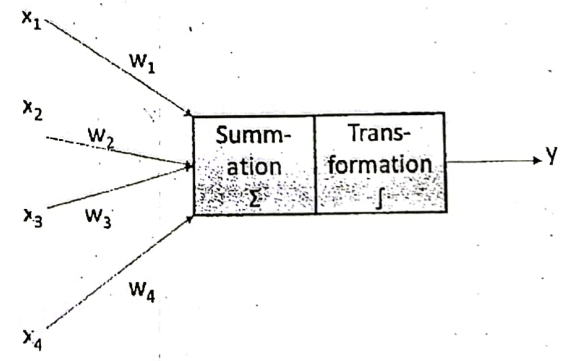
Age	Job	House	Credit	Loan Approved
Young	FALSE	No	Good	No

OR

8. a. Explain the design principles of an Artificial Neural Network. (08 Marks)

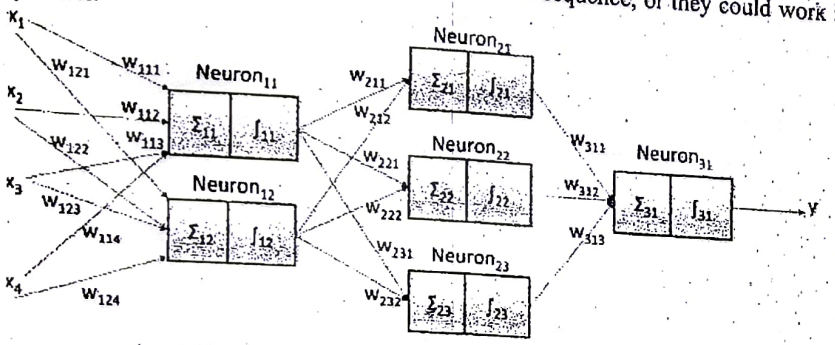
Ans. Design Principles of an ANN

1. A neuron is the basic processing unit of the network. The neuron (or processing element) receives inputs from its preceding neurons (or PEs), does some nonlinear weighted computation on the basis of those inputs, transforms the result into its output value, and then passes on the output to the next neuron in the network. X's are the inputs, w's are the weights for each input, and y is the output.



2. A neural network is a multilayered model. There is at least one input neuron, one output neuron, and at least one processing neuron. An ANN with just this basic structure would be a simple, single-stage computational unit. A simple task may be

processed by just that one neuron and the result may be communicated soon. ANNs, however, may have multiple layers of processing elements in sequence. There could be many neurons involved in a sequence depending upon the complexity of the predictive action. The layers of PEs could work in sequence, or they could work in parallel.



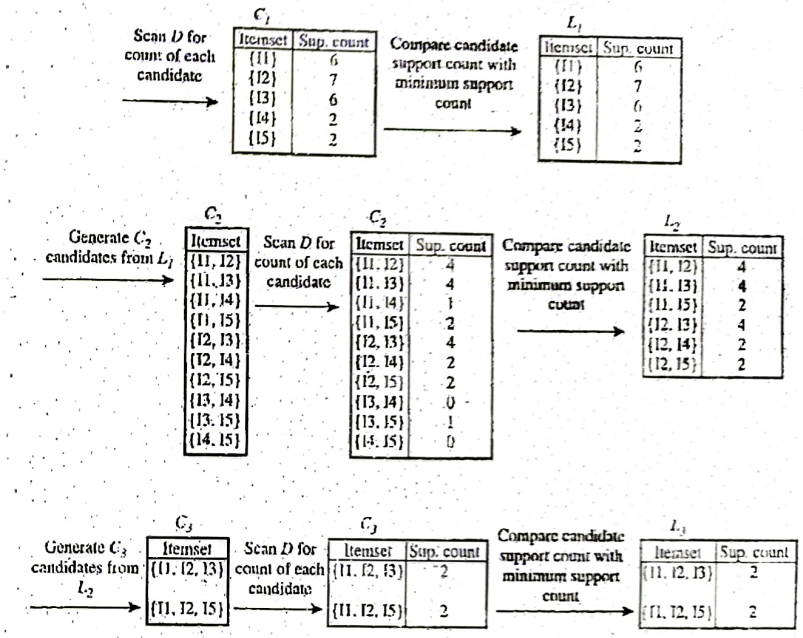
3. The processing logic of each neuron may assign different weights to the various incoming input streams. The processing logic may also use nonlinear transformation, such as a sigmoid function, from the processed values to the output value. This processing logic and the intermediate weight and processing functions are just what works for the system as a whole, in its objective of solving a problem collectively. Thus, the neural networks are considered to be an opaque and a black-box system.
4. The neural network can be trained by making similar decisions over and over again with many training cases. It will continue to learn by adjusting its internal computation and communication based on feedback about its previous decisions. Thus, the neural networks become better at making a decision as they handle more and more decisions.

b. How does the Apriori Algorithm work? Apply the same for the following example.

TID	List of Item-Ids
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

Assume the support count=2. **(08 Marks)**

Ans.



Module - 5

9. a. What is Naïve Bayes Technique? Explain its model. **(05 Marks)**

Ans. It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

Likelihood Class Prior Probability

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

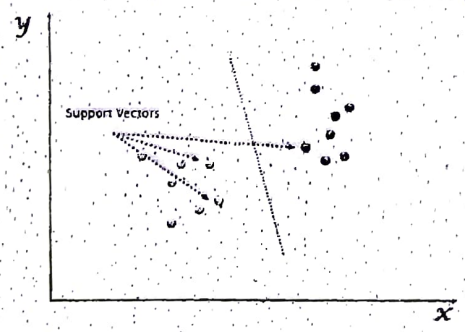
Posterior Probability Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

Above,
P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
P(c) is the prior probability of class.
P(x|c) is the likelihood which is the probability of predictor given class.
P(x) is the prior probability of predictor.

b. What is Support Vector Machine? Explain its model. **(08 Marks)**

Ans. In a nutshell, a support vector machine (or SVM) is an algorithm that works as follows. It uses a nonlinear mapping to transform the original training data into a higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane (that is, a "decision boundary" separating the tuples of one class from another). With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane. The SVM finds this hyperplane using support vectors ("essential" training tuples) and margins (defined by the support vectors).



To construct an optimal hyperplane, SVM employs an iterative training algorithm, which is used to minimize an error function. According to the form of the error function, SVM models can be classified into four distinct groups:

1. Classification SVM Type 1 (also known as C-SVM classification)
2. Classification SVM Type 2 (also known as nu-SVM classification)
3. Regression SVM Type 1 (also known as epsilon-SVM regression)
4. Regression SVM Type 2 (also known as nu-SVM regression)

Classification SVM

CLASSIFICATION SVM TYPE 1

For this type of SVM, training involves the minimization of the error function:

$$\frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$$

subject to the constraints:

$$y_i \left(w^T \phi(x_i) + b \right) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, i = 1, \ldots, N$$

where C is the capacity constant, w is the vector of coefficients, b is a constant, and represents parameters for handling nonseparable data (inputs). The index i labels the N training cases. Note that represents the class labels and xi represents the independent variables. The kernel is used to transform data from the input (independent) to the feature space. It should be noted that the larger the C, the more the error is penalized. Thus, C should be chosen with care to avoid over fitting.

CLASSIFICATION SVM TYPE 2

In contrast to Classification SVM Type 1, the Classification SVM Type 2 model minimizes the error function:

$$\frac{1}{2} w^T w - v\rho + \frac{1}{N} \sum_{i=1}^{N} \xi_i$$

subject to the constraints:

$$y_i \left(w^T \phi(x_i) + b \right) \geq \rho - \xi_i, \xi_i \geq 0, i = 1, \ldots, N \text{ and } \rho \geq 0$$

In a regression SVM, you have to estimate the functional dependence of the dependent variable y on a set of independent variables x. It assumes, like other regression problems, that the relationship between the independent and dependent variables is given by a deterministic function f plus the addition of some additive noise:

Regression SVM

y = f(x) + noise

The task is then to find a functional form for f that can correctly predict new cases that the SVM has not been presented with before. This can be achieved by training the SVM model on a sample set, i.e., training set, a process that involves, like classification (see above), the sequential optimization of an error function. Depending on the definition of this error function, two types of SVM models can be recognized:

REGRESSION SVM TYPE 1

For this type of SVM the error function is:

$$\frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i + C \sum_{i=1}^{N} \xi^* i$$

which we min mize subject to:

$$w^T \phi(x_i) + b - y_i \leq \varepsilon + \xi_i^*$$

$$y_i - w^T \phi(x_i) - b_i \leq \varepsilon + \xi_i$$

$$\xi_i, \xi_i^* \geq 0, i = 1, \ldots, N$$

REGRESSION SVM TYPE 2

For this SVM model, the error function is given by:

$$\frac{1}{2}w^T w - C\left(v\varepsilon + \frac{1}{N}\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right)\right)$$

which we minimize subject to:

$$\left(w^T\phi(x_i)+b\right) - y_i \le \varepsilon + \xi_i$$

$$y_i - \left(w^T\phi(x_i)+b_i\right) \le \varepsilon + \xi^*{}_i$$

$$\xi_i, \xi^*{}_i \ge 0, i = 1, \ldots, N, \varepsilon \ge 0$$

There are number of kernels that can be used in Support Vector Machines models. These include linear, polynomial, radial basis function (RBF) and sigmoid:

Kernel Functions

$$K(\mathbf{X}_i, \mathbf{X}_j) = \begin{cases} \mathbf{X}_i \cdot \mathbf{X}_j & \text{Linear} \\ (\gamma \mathbf{X}_i \cdot \mathbf{X}_j + C)^d & \text{Polynomial} \\ \exp\left(-\gamma |\mathbf{X}_i - \mathbf{X}_j|^2\right) & \text{RBF} \\ \tanh\left(\gamma \mathbf{X}_i \cdot \mathbf{X}_j + C\right) & \text{Sigmoid} \end{cases}$$

where $K(\mathbf{X}_i, \mathbf{X}_j) = \phi(\mathbf{X}_i) \cdot \phi(\mathbf{X}_j)$

that is, the kernel function, represents a dot product of input data points mapped into the higher dimensional feature space by transformation ϕ

Gamma is an adjustable parameter of certain kernel functions.

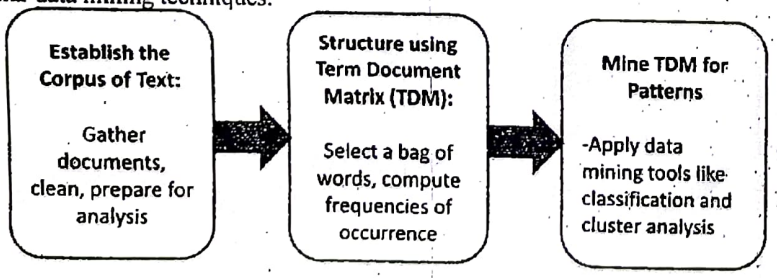
The RBF is by far the most popular choice of kernel types used in Support Vector Machines. This is mainly because of their localized and finite responses across the entire range of the real x-axis.

c. Mention the 3-steps process of Text Mining. **(03 Marks)**

Ans. Text mining is a semiautomated process. Text data needs to be gathered, structured, and then mined, in a three-step process.

1. The text and documents are first gathered into a corpus and organized.
2. The corpus is then analyzed for structure. The result is a matrix mapping important terms to source documents.
3. The structured data is then analyzed for word structures, sequences, and frequency.

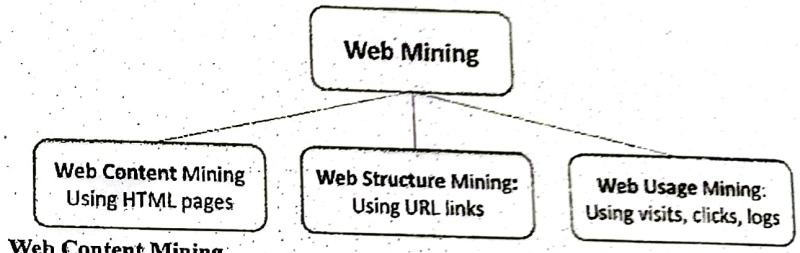
Term-document matrix (TDM): This is the heart of the structuring process. Free flowing text can be transformed into numeric data, which can then be mined using regular data mining techniques.



OR

10. a. Explain briefly the three different types of web mining. **(06 Marks)**

Ans.



Web Content Mining

A website is designed in the form of pages with a distinct URL (universal resource locator). A large website may contain thousands of pages. Those pages and their content are managed using content management systems. Every page can have text, graphics, audio, video, forms, applications, and more kinds of content, including user-generated content. The websites make a record of all requests received for its page/URLs. The log of these requests could be analyzed to gauge the popularity of those pages. The textual and application content could be analyzed for its usage by visits to the website. The pages on a website themselves could be analyzed for quality of content. The unwanted pages could be transformed with different content and style, or they may be deleted altogether. Similarly, more resources could be assigned to keep the more popular pages more fresh and inviting.

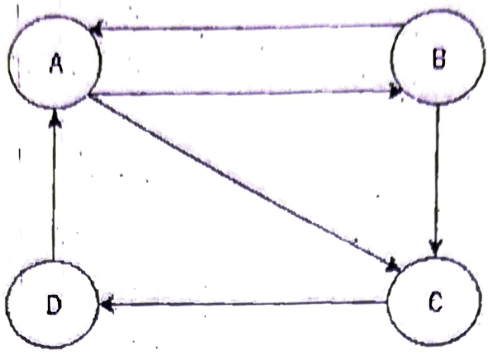
Web Structure Mining

The Web works through a system of hyperlinks using the hypertext protocol (http). Any page can create a link to any other page. The intertwined or self-referral nature of the Web lends itself to some unique analytical algorithms. The structure of web pages could also be analyzed to examine the structure of hyperlinks among pages.

Web Usage Mining

As a user clicks anywhere on a web page or application, the action is recorded by many entities in many locations. The browser at the client machine will record the click, and the web server providing the content would also log onto the pages-served activity. The entities between the client and the server, such as the router, proxy server, or ad server, too, would record that click. The goal of web usage is to extract useful information from data generated through web page visits and transactions. The activity data comes from data stored in server access logs, referrer logs, agent logs, and client-side cookies. The user characteristics and usage profiles are also gathered directly, or indirectly, through syndicated data. Further, metadata, such as page attributes, content attributes, and usage data, are also gathered.

b. Compute the rank values for the Nodes for the following network shown in FigQ10(b), Which is the Highest ranked node. Solve the same with eight iterations. **(10 Marks)**



Ans.

	Ra	Rb	Rc	Rd
Ra	0	0.50	0	1.00
Rb	0.50	0	0	0
Rc	0.50	0.50	0	0
Rd	0	0	1.00	0

Variable	Initial Value
Ra	0.250
Rb	0.250
Rc	0.250
Rd	0.250

Variable	Initial Value	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7	Iteration 8
Ra	0.250	0.375	0.313	0.344	0.328	0.336	0.332	0.334	0.333
Rb	0.250	0.125	0.188	0.156	0.172	0.164	0.168	0.167	0.167
Rc	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250
Rd	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.250

The final rank shows that rank of node A is the highest at 0.33.