PUNE DISTRICT EDUCATION ASSOCIATION'S

COLLEGE OF ENGINEERING

Manjari (BK), Pune - 412307 (Maharashtra) India

LAB MANUAL

OF

DS & BDA Lab

T E (IT) 2019

COURSE

Information Technology

Vision and Mission of Institute and Department

INSTITUTE VISION

"**Providing nurturing ground for an individual's development to make effective contribution to the society in dynamic environment through academic excellence for professional competency.**"

INSTITUTE MISSION

"**To provide the state of the art educational facilities for training students for the career in engineering and technology. To organize quality improvement programs on advances in current technology for the benefit of core stakeholders of community. To provide leadership in curriculum design and development to strengthen industry-institute commune.**"

Program Outcomes: -

PO 1. **Engineering knowledge:** An ability to apply knowledge of mathematics, including discrete mathematics, statistics, science, computer science and engineering fundamentals to model the software application.

PO 2. **Problem analysis:** An ability to design and conduct an experiment as well as interpret data, analyze complex algorithms, to produce meaningful conclusions and recommendations.

PO 3. **Design/development of solutions**:An ability to design and development of software system, component, or process to meet desired needs, within realistic constraints such as economic, environmental, social, political, health & safety, manufacturability, and sustainability.

PO 4. **Conduct investigations of complex problems**:An ability to use research based knowledge including analysis, design and development of algorithms for the solution of complex problems interpretation of data and synthesis of information to provide valid conclusion.

PO 5. **Modern tool usage:** An ability to adapt current technologies and use modern IT tools, to design, formulate, implement and evaluate computer based system, process, by considering the computing needs, limits and constraints.

PO 6. **The engineer and society:** An ability of reasoning about the contextual knowledge of the societal, health, safety, legal and cultural issues, consequent responsibilities relevant to IT practices.

PO 7. **Environment and sustainability:** An ability to understand the impact of engineering solutions in a societal context and demonstrate knowledge of and the need for sustainable development.

PO 8. **Ethics: An** ability to understand and commit to professional ethics and responsibilities and norms of IT practice.

PO 9. **Individual and team work:**An ability to apply managerial skills by working effectively as an individual, as a member of a team, or as a leader of a team in multidisciplinary projects.

PO 10. **Communication:** An ability to communicate effectively technical information in speech, presentation, and in written form

PO 11. **Project management and finance:** An ability to apply the knowledge of Information Technology and management principles and techniques to estimate time and resources needed to complete engineering project.

PO 12. **Life-long learning:** An ability to recognize the need for, and have the ability to engage in independent and life-long learning.

Experiment Learning Outcomes

At the end of Laboratory course student will be able to:

ExperimentLearningOutcome	Attainment	
	PO	**Bloomstaxonomy**
ELO 1: Able to Install Hadoop on a) Single Node b) Multiple Node and Configure Hadoop properties	PO1,PO3,PO4,PO5	4
ELO 2: Able to design and develop a distributed application using MapReduce to processes a log file of a system.	PO1,PO2,PO3	4
ELO 3: Able to Write an application using HiveQL for flight information system which will include operation.	PO1,PO2,PO3	4
ELO 4: Able to Perform the following operations using Python on the Facebook metrics data sets	PO1,PO3,PO4	4
ELO 5: Able to Perform the following operations using Python on the Air quality and Heart Diseases data sets	PO1,PO2,PO3,	4
ELO6:Able to Integrate Python and Hadoop and perform the following operations on forest fire dataset	PO1,PO2,PO3,	4
ELO 7: Able to Visualize the data using Python libraries matplotlib,seaborn by plotting the graphs for assignment	PO1,PO2,PO3,	4
ELO 8: Able to Perform the following data visualization operations using Tableau on Adult and Iris datasets.	PO1,PO2,PO3,	4

ELO 9: Able To Create a review scrapper for any ecommerce website to fetch real time comments, reviews, ratings, comment tags, customer name using Python.	PO1,PO2,PO3,	4
ELO 10:To Develop a mini project in a group using different predictive models techniques to solve any real life problem.	PO1	2

314457: DS & BDA Lab

Part A : Assignments based on the Hadoop

Practical Session Plan

Time (min)	Content	Learning Aid / Methodology	Faculty Approach	Typical Student Activity	Skill / Competency Developed

05	Relevance and significance of experiment	Chalk & Talk , Presentation	Introduces, Facilitates, Monitors	Listens, Participates, Discusses	Knowledge, Communication, intrapersonal
10	Explanation of experiment	Chalk & Talk , Presentation	Introduces, Facilitates, Explains	Listens	Knowledge, Communication, intrapersonal, Application
90	Designing & Coding	demonstration	Explains, Monitors	Participates, Discusses	comprehension, Hands on experiment
10	Testing	Evaluation	Explains, Monitors	Participates, Discusses	Knowledge, Communication, Intrapersonal, Application
05	Output and conclusions	Demonstration Presentation	Lists, Facilitates	Listens, Participates, Discusses	Knowledge, Communication, intrapersonal, Comprehension

Title: To perform Single node/Multiple node Hadoop Installation.

Objective: To study,

1. Configure Hadoop on open source software

ELO1: Able to install Hadoop on Single Node Cluster

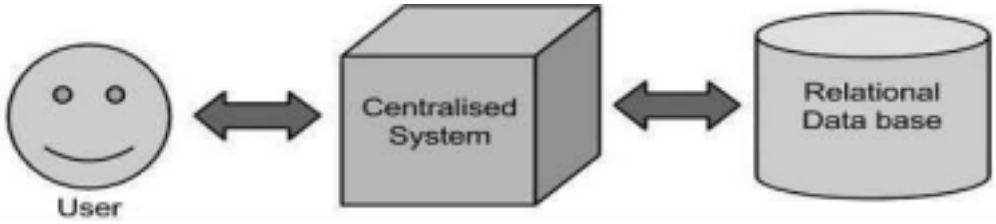
- Theory:

Hadoop

Hadoop is an open source software framework written in Java or distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures (of individual machines or racks of machines) are common and thus should be automatically handled in software by the framework.

Traditional Approach

In this approach, an enterprise will have a computer to store and process big data. Here data will be stored in an RDBMS like Oracle Database, MS SQL Server or DB2 and sophisticated software can be written to interact with the database, process the required data and present it to the users for analysis purpose.

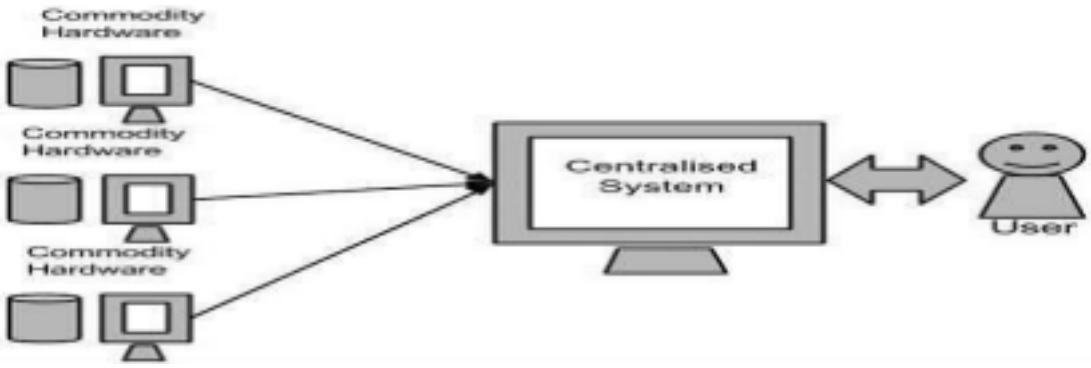


Limitation

This approach works well where we have less volume of data that can be accommodated by standard database servers, or up to the limit of the processor which is processing the data. But when it comes to dealing with huge amounts of data, it is really a tedious task to process such data through a traditional database server.

Google's Solution

Google solved this problem using an algorithm called MapReduce. This algorithm divides the task into small parts and assigns those parts to many computers connected over the network, and collects the results to form the final result dataset.

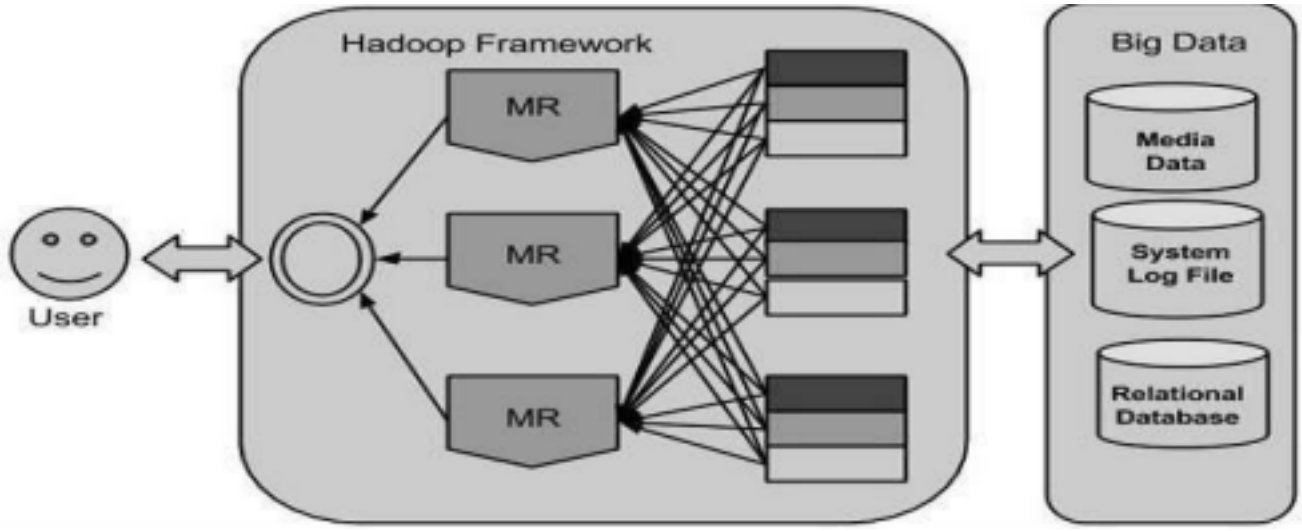


Above diagram shows various commodity hardware's which could be single CPU machines or servers with higher capacity.

Hadoop

Doug Cutting, Mike Cafarella and team took the solution provided by Google and started an Open Source Project called HADOOP in 2005 and Doug named it after his son's toy elephant. Now Apache Hadoop is a registered trademark of the Apache Software Foundation.

Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel on different CPU nodes. In short, Hadoop framework is capable enough to develop applications capable of running on clusters of computers and they could perform complete statistical analysis for huge amounts of data.



Hadoop Architecture

Hadoop framework includes following four modules:

Hadoop Common: These are Java libraries and utilities required by other Hadoop modules. These libraries provides filesystem and OS level abstractions and contains the necessary Java files and

scripts required to start Hadoop.

Hadoop YARN: This is a framework for job scheduling and cluster resource management.

Hadoop Distributed File System (HDFS™): A distributed file system that provides high throughput access to application data.

Hadoop MapReduce: This is YARN-based system for parallel processing of large data sets.

MapReduce

Hadoop **MapReduce** is a software framework for easily writing applications which process big amounts of data in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

The term MapReduce actually refers to the following two different tasks that Hadoop programs perform:

- **The Map Task:** This is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs).
- **The Reduce Task:** This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.

Typically both the input and the output are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for resource management, tracking resource consumption/availability and scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves TaskTracker execute the tasks as directed by the master and provide task-status information to the master periodically.

The JobTracker is a single point of failure for the Hadoop MapReduce service which means if JobTracker goes down, all running jobs are halted.

Hadoop Distributed File System

Hadoop can work directly with any mountable distributed file system such as Local FS, HFTP FS, S3 FS, and others, but the most common file system used by Hadoop is the Hadoop Distributed File System (HDFS).

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters (thousands of computers) of small computer machines in a reliable, fault-tolerant manner.

HDFS uses a master/slave architecture where master consists of a single **NameNode** that manages thefile system metadata and one or more slave **DataNodes** that store the actual data.

How to Install Hadoop on Ubuntu 18.04 or 20.04

1.Install OpenJDK on Ubuntu.

2. Set Up a Non-Root User for Hadoop Environment. Install Open SSH onUbuntu. ...

3. Download and Install Hadoop on Ubuntu.

4. Single Node Hadoop Deployment (Pseudo-Distributed Mode) ...

5. Format HDFS NameNode.

6. Start Hadoop Cluster.

7. Access Hadoop UI from Browser.

How Does Hadoop Work?

Stage 1

A user/application can submit a job to the Hadoop (a hadoop job client) for required process by specifying the following items:

1. The location of the input and output files in the distributed file system.
2. The java classes in the form of jar file containing the implementation of map and reduce functions.
3. The job configuration by setting different parameters specific to the job.

Stage 2

The Hadoop job client then submits the job (jar/executable etc) and configuration to the JobTracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

Stage 3

The TaskTrackers on different nodes execute the task as per MapReduce implementation and output of the reduce function is stored into the output files on the file system.

Advantages of Hadoop

- Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores.
- Hadoop does not rely on hardware to provide fault-tolerance and high availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.

a) Single Node:

Steps for Compilation & Execution

- sudo apt-get update
- sudo apt-get install openjdk-8-jre-headless

- sudo apt-get install openjdk-8-jdk
- sudo apt-get install ssh
- sudo apt-get install rsync
- # Download hadoop from : http://www.eu.apache.org/dist/hadoop/common/stable/hadoop 2.7.1.tar.gz
- # copy and extract hadoop-2.7.1.tar.gz in home folder
- # rename the name of the extracted folder from hadoop-2.7.1 to hadoop •

readlink -f /usr/bin/javac

- gedit ~/hadoop/etc/hadoop/hadoop-env.sh
- # add following line in it
- **# for 32 bit ubuntu**

- export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-i386
- **# for 64 bit ubuntu**
- export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
- # save and exit the file
- # to display the usage documentation for the hadoop script try next command •

~/hadoop/bin/hadoop

 # Pseudo-Distributed mode

- # get your user name
- whoami
- # remember your user name, we'll use it in the next step
- gedit ~/hadoop/etc/hadoop/core-site.xml

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:1234</value>
    </property>
</configuration>
```

- gedit ~/hadoop/etc/hadoop/hdfs-site.xml

```
<configuration>
```



```
    <property>
        <name>dfs.replication</name>
        <value>1</value>
        </property>
        <property>
        <name>dfs.name.dir</name>
        <value>file:///home/your_user_name/hadoop/name_dir</value>
        </property>
        <property>
        <name>dfs.data.dir</name>
```

<value>file:///home/**your_user_name**/hadoop/data_dir</value>

</property>

</configuration>

#Setup pass phrase less/passwordlessssh

- ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
- cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
- export HADOOP_PREFIX=/home/**your_user_name**/hadoop
- ssh localhost

type **exit** in the terminal to close the ssh connection (very important)

Exit

 # The following instructions are to run a MapReduce job locally.

- **Format the filesystem:(Do it only once)**

  ~/hadoop/bin/hdfsnamenode -format

- **Start NameNode daemon and DataNode daemon:**

  ~/hadoop/sbin/start-dfs.sh

- **Browse the web interface for the NameNode; by default it is available at:**

  http://localhost:50070/

- **Make the HDFS directories required to execute MapReduce jobs:**

  ~/hadoop/bin/hdfsdfs -mkdir /user

  ~/hadoop/bin/hdfsdfs -mkdir /user/**your_user_name**

- **Copy the sample files (from ~/hadoop/etc/hadoop) into the distributed filesystem folder(input)**

  ~/hadoop/bin/hdfsdfs -put ~/hadoop/etc/hadoop input

- **Run the example map-reduce job**

  ~/hadoop/bin/hadoop jar ~/hadoop/share/hadoop/mapreduce/hadoop-mapreduce examples-2.7.1.jar grep input output 'us[a-z.]+'

- **View the output files on the distributed filesystem**

  ~/hadoop/bin/hdfsdfs -cat output/*

- **Copy the output files from the distributed filesystem to the local filesystem and examine**

them:

~/hadoop/bin/hdfsdfs -get output output

- **remove local output folder**

rm -r output

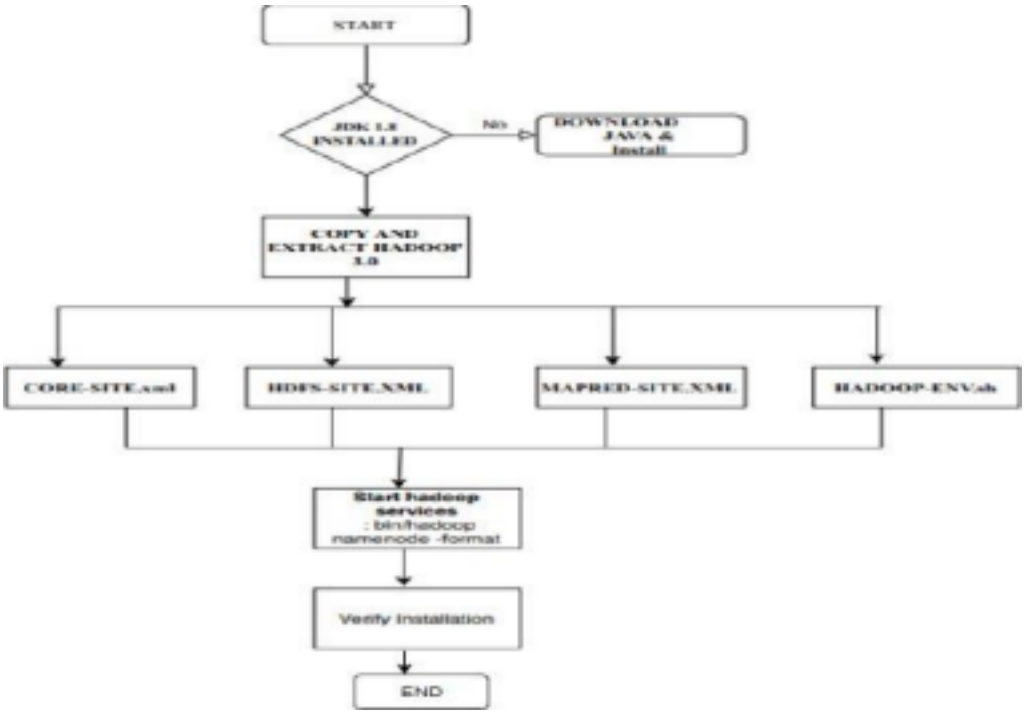
- **remove distributed folders (input & output)**

~/hadoop/bin/hdfsdfs -rm -r input output

- **When you're done, stop the daemons with**

~/hadoop/sbin/stop-dfs.sh

- **Flow Chart**



Reference :

- https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster
- https://hadoop.apache.org/docs/r2.7.6/hadoop-project-dist/hadoop common/SingleCluster.html

Software Requirement:

1. Ubuntu 18 / 18
2. Hadoop2.7.1

Conclusion: In this way the Hadoop was installed & configured on Ubuntu for BigData.

Questions:

Q1) What are the various daemons in Hadoop and their role in Hadoop cluster? Q2) What does JPS command do?

Q3) What is difference between RDBMS&.Hadoop

Q4) What is YARN and explain its components?

Q5) Explain HDFS and its components?

EXPERIMENT NO.2

Title:

Design a distributed application using MapReduce(Using Java) which processes a log file of a system. List out the users who have logged for maximum period on the system. Use simple log file from the Internet and process it using a pseudo distribution mode on Hadoop platform.

Objectives: To learn the concept of Mapper and Reducer and implement it for log file processing

Aim: To implement a MapReduce program that will process a log file of a system.

Theory

- Introduction

MapReduce is a framework using which we can write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner.MapReduce is a processing technique and a program model for distributed computing based on java.

The MapReduce algorithm contains two important tasks, namely Map and Reduce.Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).

Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

Under the MapReduce model, the data processing primitives are called mappers and reducers. once we writean application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change

Algorithm

- MapReduce program executes in three stages, namely **map stage**, **shuffle stage**, and **reduce stage**.
    - Input : file or directory
    - Output : Sorted file**<key, value>**

1. Mapstage :

- The map or mapper's job is to process the input data.
- Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS).
- The input file is passed to the mapper function line by line.
- The mapper processes the data and creates several small chunks of data.

2. Shuffle stage:

- This phase consumes the output of mapping phase.
- Its task is to consolidate the relevant records from Mapping phase output

3. Reduce stage :

- This stage is the combination of the Shuffle stage and the Reduce stage.
- The Reducer's job is to process the data that comes from the mapper.
- After processing, it produces a new set of output, which will be stored in theHDFS.

Inserting Data into HDFS:

•The MapReduce framework operates 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 as the output of the job, conceivably of different types.

•The key and the value classes should be in serialized manner by the framework and hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable Comparable interface to facilitate sorting by the framework.

•Input and Output types of a MapReduce job: **(Input) <k1, v1> -> map -><k2, v2>-> reduce -><k3, v3> (Output).**

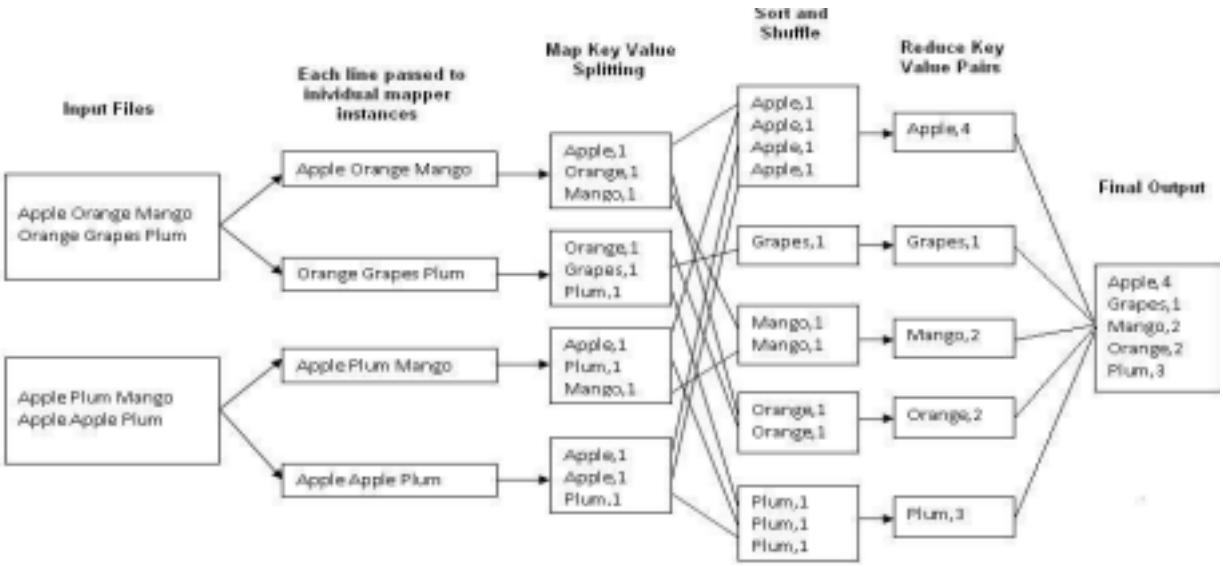
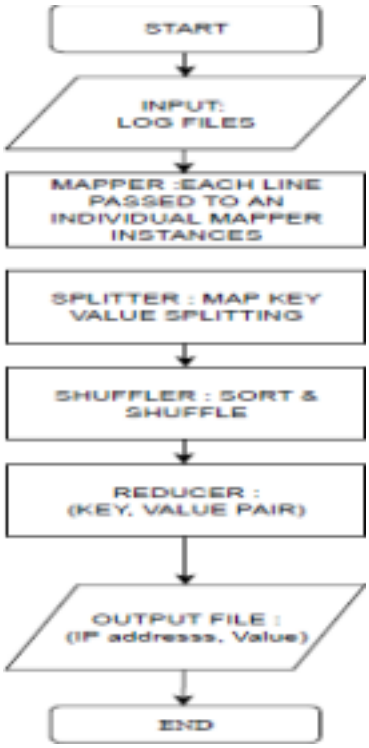


Fig.2.1 : An Example Program to Understand working of MapReduce Program.

Flow chart



Program Code

#Mapper Class

```
packageSalesCountry;

importjava.io.IOException;

importorg.apache.hadoop.io.IntWritable;
importorg.apache.hadoop.io.LongWritable;
importorg.apache.hadoop.io.Text;
importorg.apache.hadoop.mapred.*;

public class SalesMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text,
IntWritable> {
        private final static IntWritable one = new IntWritable(1);

        public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output,
Reporter reporter) throws IOException {

                String valueString = value.toString();
                String[] SingleCountryData = valueString.split("-");
                output.collect(new Text(SingleCountryData[0]), one);
        }
}
```

#Reducer Class

```
packageSalesCountry;

importjava.io.IOException;
importjava.util.*;

importorg.apache.hadoop.io.IntWritable;
importorg.apache.hadoop.io.Text;
importorg.apache.hadoop.mapred.*;

public class SalesCountryReducer extends MapReduceBase implements Reducer<Text, IntWritable,
Text, IntWritable> {

        public void reduce(Text t_key, Iterator<IntWritable> values, OutputCollector<Text,IntWritable>
output, Reporter reporter) throws IOException {
                Text key = t_key;
                intfrequencyForCountry = 0;
                while (values.hasNext()) {
                        // replace type of value with the actual type of our value
                        IntWritable value = (IntWritable) values.next();
                        frequencyForCountry += value.get();

                }
                output.collect(key, new IntWritable(frequencyForCountry));
```



```
        }
}
```

#Driver Class

```
packageSalesCountry;

importorg.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
importorg.apache.hadoop.mapred.*;

public class SalesCountryDriver {
        public static void main(String[] args) {
                JobClientmy_client = new JobClient();
                // Create a configuration object for the job
                JobConfjob_conf = new JobConf(SalesCountryDriver.class);

                // Set a name of the Job
                job_conf.setJobName("SalePerCountry");

                // Specify data type of output key and value
                job_conf.setOutputKeyClass(Text.class);
                job_conf.setOutputValueClass(IntWritable.class);

                // Specify names of Mapper and Reducer Class
                job_conf.setMapperClass(SalesCountry.SalesMapper.class);
                job_conf.setReducerClass(SalesCountry.SalesCountryReducer.class);

                // Specify formats of the data type of Input and output
                job_conf.setInputFormat(TextInputFormat.class);
                job_conf.setOutputFormat(TextOutputFormat.class);

                // Set input and output directories using command line arguments,
                //arg[0] = name of input directory on HDFS, and arg[1] = name of output directory
to be created to store the output file.

                FileInputFormat.setInputPaths(job_conf, new Path(args[0]));
                FileOutputFormat.setOutputPath(job_conf, new Path(args[1]));

                my_client.setConf(job_conf);
                try {
                        // Run the job
                        JobClient.runJob(job_conf);
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }
}
```

Steps for Compilation & Execution of Program:

#sudomkdiranalyzelogs

ls

#sudochmod -R 777 analyzelogs/

cd

ls

cd ..

pwd

ls

cd

pwd

#sudochown -R hduseranalyzelogs/

cd

ls

#cd analyzelogs/

ls

cd ..

Copy the Files (Mapper.java,Reduce.java,Driver.java to Analyzelogs Folder)

#sudocp /home/mde/Desktop/count_logged_users/* -/analyzelogs/

Start HADOOP

#start-dfs.sh

#start-yarn.sh

#jps

cd

cdanalyzelogs

ls

pwd

ls

#ls -ltr

#ls -al

#sudochmod +r *.*

pwd

#export CLASSPATH="$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce client-core-2.9.0.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-client common-2.9.0.jar:$HADOOP_HOME/share/hadoop/common/hadoop-common 2.9.0.jar:~/analyzelogs/SalesCountry/*:$HADOOP_HOME/lib/*"

Compile Java Files

javac -d . SalesMapper.java SalesCountryReducer.java SalesCountryDriver.java

ls

#cd SalesCountry/

ls

cd ..

#sudogedit Manifest.txt

#jar -cfm analyzelogs.jar Manifest.txt SalesCountry/*.class

ls

cd

jps

#cd analyzelogs/

Create Directory on Hadoop

#sudomkdir ~/input2000

ls

pwd

#sudocp access_log_short.csv ~/input2000/

$HADOOP_HOME/bin/hdfsdfs -put ~/input2000 /

$HADOOP_HOME/bin/hadoop jar analyzelogs.jar /input2000 /output2000 # $HADOOP_HOME/bin/hdfsdfs -cat /output2000/part-00000

 # stop-all.sh

jps

Output:

hduser@com17-Veriton-M200-A780:~/analyzelog$ $HADOOP_HOME/bin/hdfsdfs -cat /output2000/part-00000

18/01/08 10:13:25 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
10.1.1.236 7
10.1.181.142 14
10.1.232.31 5
10.10.55.142 14
10.102.101.66 1
10.103.184.104 1
10.103.190.81 53
10.103.63.29 1
10.104.73.51 1
10.105.160.183 1
10.108.91.151 1
10.109.21.76 1
10.11.131.40 1
10.111.71.20 8
10.112.227.184 6
10.114.74.30 1
10.115.118.78 1
10.117.224.230 1
10.117.76.22 12
10.118.19.97 1
10.118.250.30 7
10.119.117.132 23
10.119.33.245 1
10.119.74.120 1
10.12.113.198 2
10.12.219.30 1
10.120.165.113 1
10.120.207.127 4
10.123.124.47 1
10.123.35.235 1
10.124.148.99 1
10.124.155.234 1
10.126.161.13 7
10.127.162.239 1
10.128.11.75 10
10.13.42.232 1
10.130.195.163 8
10.130.70.80 1
10.131.163.73 1
10.131.209.116 5
10.132.19.125 2
10.133.222.184 12

10.134.110.196 13
10.134.242.87 1
10.136.84.60 5
10.14.2.86 8
10.14.4.151 2

hduser@com17-Veriton-M200-A780:~/analyzelog$

Conclusion: Thus we have learnt how to design a distributed application using MapReduce and process a log file of a system.

EXPERIMENT NO. 3

Part A: Assignments based on the Hadoop HBase via Hive

Title:

Write an application using HiveQL for flight information system which will include

a. Creating, Dropping, and altering Database tables.

b. Creating an external Hive table.

c. Load table with data, insert new values and field in the table, Join tables with Hive

d. Create index on Flight Information Table

e. Find the average departure delay per day in 2008.

Objectives: 1) To describe the basics of Hive

2)Explain the components of the Hadoop ecosystem

Aim: To execute a Hive, HBase Query that will perform CRUD operation on Flight Table

Theory

Hive – Introduction

Hive is defined as a data warehouse system for Hadoop that facilitates ad-hoc queries and the analysis of large datasets stored in Hadoop.

Following are the facts related to Hive:

- It provides a SQL-like language called **HiveQL(HQL).** Due to its SQL-like interface, Hive is a popular choice for Hadoop analytics.
- It provides massive scale-out and faults tolerance capabilities for data storage and processing of commodity hardware.
- Relying on MapReduce for execution, Hive is batch-oriented and has high latency for query execution

Hive – Characteristics

- Hive is a system for managing and querying unstructured data into a structured format.

- It uses the concept of MapReduce for the execution of its scripts and the Hadoop Distributed File System or HDFS for storage and retrieval of data.

Following are the **key principles** underlying Hive.

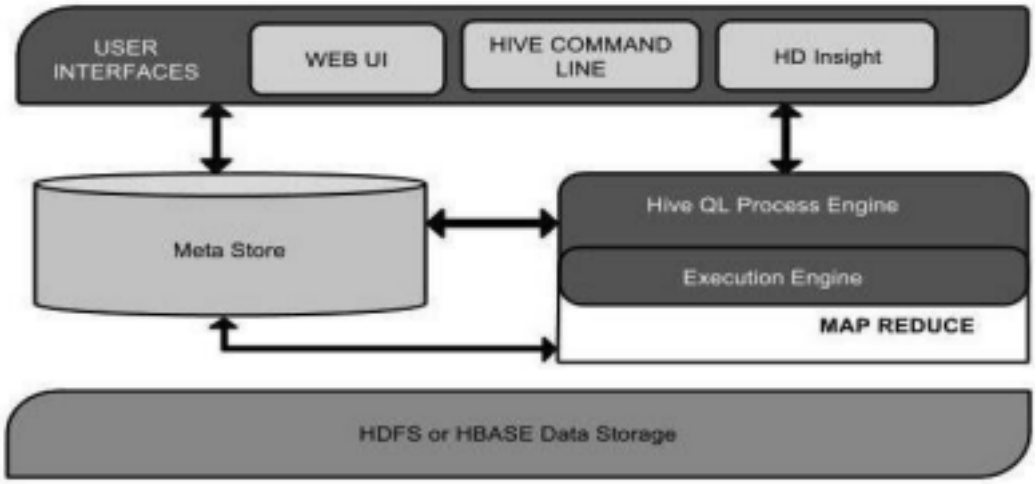
• Hive commands are similar to that of SQL. SQL is a data warehousing tool that is similar to Hive.

• Hive contains extensive, pluggable MapReduce scripts in the language of your choice. These scripts include rich, user-defined data types and user-defined functions.

• Hive has an extensible framework to support different files and data formats. • Performance is better in Hive since Hive engine uses the best-inbuilt script to reduce the execution time, thus enabling high output in less time.

Architecture of Hive

The following component diagram depicts the architecture of Hive:



Unit Name	Operation
User Interface	Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. The user interfaces that Hive supports are Hive Web UI, Hive command line, and Hive HD Insight
Meta Store	Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.

HiveQL Process Engine	HiveQL is similar to SQL for querying on schema info on the Metastore. Instead of writing MapReduce program in

	Java, we can write a query for MapReduce job and process it.
Execution Engine	The conjunction part of HiveQL process Engine and MapReduce is Hive Execution Engine. Execution engine processesthe query and generates results as same as MapReduce results.
HDFS or HBASE	Hadoop distributed file system or HBASE are the data storage techniques to store data into file system.

HBase – Introduction

• Apache HBase is a distributed, column-oriented database built on top of HDFS. • Apache HBase can scale horizontally to thousands of commodity servers and petabytes of data by indexing the storage.

• Apache HBase is an open-source, distributed, and versioned non-relational database modeled after Google's Bigtable: A Distributed Storage System for Structured Data.

• The goal of HBase is to host very large tables with billions of rows and millions of columns, atop clusters of commodity hardware.

Characteristics of HBase

HBase is a type of NoSQL database and is classified as a key-value store.In HBase,

• Value is identified with a key.

• Both key and value are byte-array, which means binary formats can be stored easily. • Values are stored in key-orders and can be accessed quickly by their keys. • HBase is a database in which tables have no schema. Column families and not columns are defined at the time of table creation.

HBase Architecture

HBase has two types of Nodes which are Master and RegionServer.

Master

• There is only one Master node running at a time whereas there can be one or more RegionServers. The high availability of the Master node is maintained with

ZooKeeper.

- The Master node manages cluster operations like an assignment, load balancing, and splitting. It is not a part of read or write path.

RegionServer

- The RegionServer hoststables, performs reads, and buffers writes. Clients communicate with RegionServer to read and write.
- A region in HBase is the subset of a table's rows. The Master node detects the status of RegionServers and assigns regions to RegionServers.

Commands for HBase

StartingHBase

./bin/hbase shell

Create Table Statement

hbase(main):002:0> create 'emp', 'personal data', 'professional data'

Verification

hbase(main):002:0> list

Disabling a Table using HBase

hbase(main):025:0> disable 'emp'

Enabling a Table using HBase

hbase(main):005:0> enable 'emp'

describe

hbase> describe 'table name'

alter

hbase> alter 't1', NAME ⇒ 'f1', VERSIONS ⇒ 5

Dropping a Table

hbase(main):018:0> disable 'emp'

Stopping HBase

./bin/stop-hbase.sh

Listing 13-1: Installing Apache Hadoop and Hive

$ mkdirhadoop; cp hadoop-1.2.1.tar.gz hadoop; cd hadoop

$ gunzip hadoop-1.2.1.tar.gz

$ tar xvf *.tar

$ mkdir hive; cp hive-0.11.0.tar.gz hive; cd hive

$ gunzip hive-0.11.0.tar.gz

$ tar xvf *.tar

Listing 13-2: Setting Up Apache Hive Environment Variables in .bashrc export HADOOP_HOME=/home/user/Hive/hadoop/hadoop-1.2.1 export JAVA_HOME=/opt/jdk export HIVE_HOME=/home/user/Hive/hive-0.11.0 export PATH=$HADOOP_HOME/bin:$HIVE_HOME/bin: $JAVA_HOME/bin:$PATH

Listing 13-3: Setting Up the hive-site.xml File

$ cd $HIVE_HOME/conf

$ cp hive-default.xml.templateto hive-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
    <!-- Hive Execution Parameters -->
    <property>
    <name>hive.metastore.warehouse.dir</name>
    <value>/home/biadmin/Hive/warehouse</value>
    <description>location of default database for the
    warehouse</description> </property>
</configuration>
```

Working with Hive Data Types

Listing 13-7 goes to the trouble of creating a table that uses all Hive-supported data types and the amount of memory required.

Commands for Hive

Create Database Statement In Hive

hive> CREATE DATABASE userdb;

Create Table Statement

hive> CREATE TABLE IF NOT EXISTS employee (eidint, name String,

salary String, destination String)COMMENT 'Employee details'ROW

FORMAT DELIMITED FIELDS TERMINATED BY '\t'LINES

TERMINATED BY '\n'STORED AS TEXTFILE;

Alter Table Statement

hive> ALTER TABLE employee RENAME TO emp;

Drop Table Statement

hive>DROP TABLE IF EXISTS employee;

1) Creating, Dropping, and altering Database tables

hbase(main):001:0> create 'flight','finfo','fsch'

SLF4J: Class path contains multiple SLF4J bindings.

SLF4J: Found binding in [jar:file:/home/suruchi/hbase/lib/slf4j log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: Found binding in [jar:file:/usr/local/hadoop-1.2.1/lib/slf4j log4j12-1.4.3.jar!/org/slf4j/impl/StaticLoggerBinder.class]

SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.

0 row(s) in 1.6440 seconds
=>Hbase::Table – flight

hbase(main):014:0> disable'tb1'

0 row(s) in 1.4940 seconds

hbase(main):015:0> drop 'tb1'

0 row(s) in 0.2540 seconds

2) Load table with data, insert new values and field in the table, Join tables with Hive

hbase(main):002:0> put 'flight',1,'finfo:dest','mumbai'

0 row(s) in 0.1400 seconds

hbase(main):003:0> put 'flight',1,'finfo:source','pune'

0 row(s) in 0.0070 seconds

hbase(main):004:0> put 'flight',1,'fsch:at','10.25am'

0 row(s) in 0.0120 seconds

hbase(main):005:0> put 'flight',1,'fsch:dt','11.25am'

0 row(s) in 0.0100 seconds

hbase(main):006:0> scan 'flight'

ROW COLUMN+CELL

1 column=finfo:dest, timestamp=1554629442188, value=mumbai 1

column=finfo:source, timestamp=1554629455512, value=pune

1 column=fsch:at, timestamp=1554629478320, value=10.25am 1
column=fsch:dt, timestamp=1554629491414, value=11.25am

1 row(s) in 0.0450 seconds

hbase(main):007:0> alter 'flight',Name='revenue'

Updating all regions with the new schema...

0/1 regions updated.

1/1 regions updated.

Done.

0 row(s) in 2.3720 seconds

hbase(main):008:0> put 'flight',1,'revenue',10000

0 row(s) in 0.0110 seconds

hbase(main):016:0> get 'flight',1

COLUMN CELL

finfo:dest timestamp=1554629442188, value=mumbai

finfo:source timestamp=1554629455512, value=pune

fsch:at timestamp=1554629478320, value=10.25am

fsch:dt timestamp=1554629491414, value=11.25am

revenue: timestamp=1554629582539, value=10000

5 row(s) in 0.0310 seconds

3) Creating an external Hive table to connect to the HBase for Customer Information Table

```
hive> CREATE EXTERNAL TABLE
FLIGHT_1(Row_keystring,sourcestring,deststring,atstring,dt string)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH
SERDEPROPERTIES("hbase.columns.mapping"=":key,finfo:source,finfo:de
st,fsch:at,fsch:to") TBLPROPERTIES("hbase.table.name"="flight");
```

4) Create index on Flight information Table

```
hive>CREATE INDEX ine ON TABLE FLIGHT(source) AS
'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler' WITH
DEFERRED REBUILD;

OK
Time taken: 1.841 seconds
hive>SHOW INDEX ON FLIGHT;

OK
```

Time taken: 0.126 seconds, Fetched: 1 row(s)

5) Find the average departure delay per day in 2008.

hive>select avg(delay) from flight where year = 2008;

Conclusion: Thus we have learnt to integrate Hive with Hbase

Part-B

EXPERIMENT NO.4

1. Perform the following operations using Python on the Facebook metrics data sets

a. Create data subsets

b. Merge Data

c. Sort Data

d. Transposing Data

e. Shape and reshape Data

Objectives:

1.To understand and apply the Analytical concept of Big data using R/Python. 2. To study detailed concept R.

Aim: To perform basic analytical operation on given dataset

Theory:

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language, including NASA, Google, YouTube, BitTorrent, etc.

Features of Python is as it is a dynamic, high level, free open source and interpreted programming language. It supports object-oriented programming as well as procedural oriented programming.

1. Easy to code:

Python is a high-level programming language. Python is very easy to learn the language as

compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer friendly language.

2. Free and Open Source:

Python language is freely available at the official website and you can download it from the given download link

Group B:Assignment based on Data analytic using python


```
import pandas as pd
import numpy as np
df=pd.read_excel("dataset_Facebook_1_xlsx")
```

#Perform following operation using python on facebook metrics data sets

```
df.head()
df.info()
df.isnull()
df.dropna(how='any',axis=0)
```

Create data subsets

```
df1=df.loc[1:245,['Category','Lifetime Post Total Reach','Type','Total Interactions']]
df2=df.loc[245:500,['Post Month','Post Weekday','Post Hour','Lifetime Post
Consumers']] df1
df2
```

Merge 2 dataset/subsets

```
df_row = pd.concat([df1, df2])
df_row
```

#shape and reshape data

df.shape

df.melt()

Transposing Data

df.transpose()

df1.transpose()

df2.transpose()

Sorting data

df.sort_values(by='Category')

df.sort_index()

CONCLUSION: Thus we have learnt how to perform the different reshape operations using python**. (ALL students are expected to put the screen shot of their execution)**

Part-B

EXPERIMENT NO. 5

Title:

Perform the following operations using Python on the Air quality and Heart Diseases data sets 1) Data cleaning 2) Data integration

3) Data transformation 4) Error correcting

5) Data model building

Objectives:

1.To understand and apply the Analytical concept of Big data using Python. 2.To study detailed concept Python.

THEORY:

Data cleaning or data preparation is an essential part of statistical analysis. In fact ,in practice itis often more time-consuming than the statistical analysis itself

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import random as rd

ds = pd.read_excel("AirQuality.xlsx")

ds_heart = pd.read_csv("heart.csv")

ds.head()

ds.info()
```

<class 'pandas.core.frame.DataFrame'> RangeIndex: 9357 entries, 0 to 9356 Data columns (total 15 columns): # Column Non-Null Count Dtype ---------------------------- 0 Date 9357 non-null object 1 Time 9357 non-null object 2 CO(GT) 9357 non-null object 3 PT08.S1(CO) 9357 non-null int64 4 NMHC(GT) 9357 non-null int64


```
ds.isnull().sum()
ds.dropna()
```

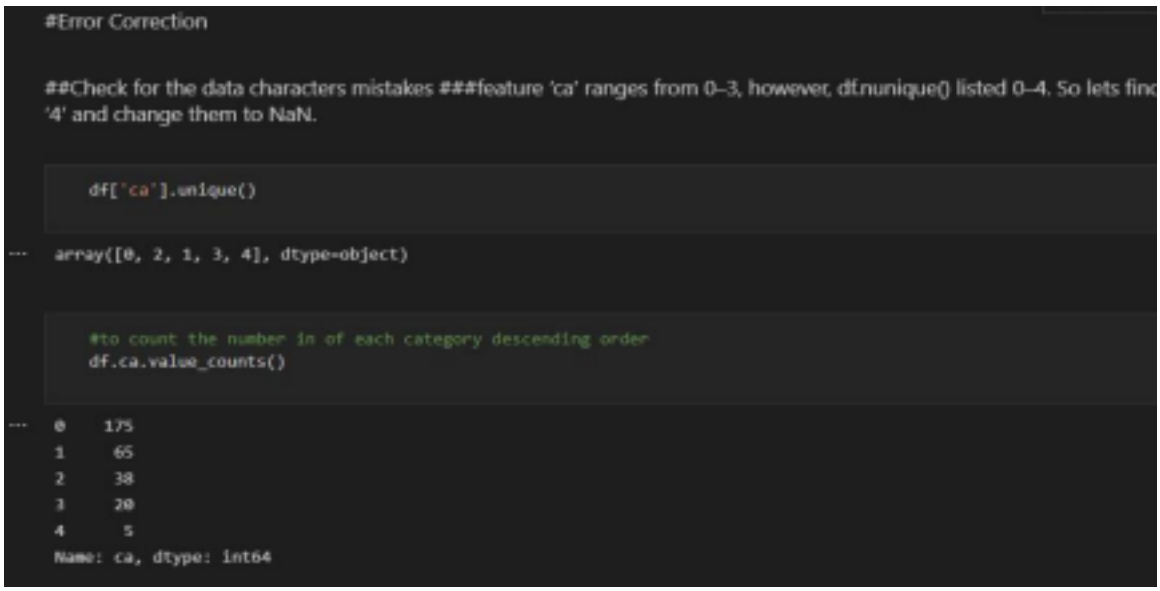
2) Data integration

```
ds1 = ds.loc[111:999, ['Date', 'Time', 'C6H6(GT)', 'RH']]
ds2 = ds.iloc[[1,3,5,2,4,22,43,54,67,7,8,9,50,10,11]]

ds_integration = pd.concat([ds1,ds2])

ds_integration
```

3) Data transformation

```
ds_integration.transpose()


ds.drop(columns = "NOx(GT)")
ds2.drop(1)
ds.melt()


ds_merged = pd.concat([ds,ds_heart])
ds_merged
```

4)Error correcting:



```
#Error Correction

##Check for the data characters mistakes ###feature 'ca' ranges from 0–3, however, df.nunique() listed 0–4. So lets find
'4' and change them to NaN.

    df['ca'].unique()

array([0, 2, 1, 3, 4], dtype=object)


    #to count the number in of each category descending order
    df.ca.value_counts()

0    175
1     65
2     38
3     20
4      5
Name: ca, dtype: int64
```

5) Data model building

 # Step1 : Divide the dataset into training and Testing

```
library(caTools)

hdata[, c(1)] <- sapply(hdata[, c(1)], as.numeric)

set.seed(123)

split = sample.split(hdata$num, SplitRatio = 2/3)

train_hdata = subset(hdata, split == TRUE)

test_hdata = subset(hdata, split == FALSE)

#You can use following code for creating training and testing samples,

# train_hdata=hdata[1:212,]

# test_hdata=hdata[213:303,]
```

```
dim(train_hdata)
#[1] 212 14
dim(test_hdata)
#[1] 91 14
```

 # Step 2: Use prediction Model using any of the technique-like regression, Classification and clustering

```
# here I have used Technique 1-Linear regression, 2-Multiple regression, 3-kNN,
4-Naive Bayes technique for prediction
```

 # Technique 1: Linear regression

```
# Here for hear disease dataset, Variable age is IV and num is IV for linear
regression model
# fitting simple linear Regression to the training set
library(caTools)
regressor=lm(formula = num~age, data=train_hdata)
#predicting the test set result using regressor
hd_age_predict=predict(regressor, newdata=test_hdata)
# As the result is not whole number, rounding the result
round_age=hd_age_predict
rage=round(round_age)
# Displaying the accuracy using confusion Matrix
library(e1071)
library(caret)
df=confusionMatrix(rage,test_hdata$num)
```



```
# Confusion Matrix and Statistics
# Reference
# Prediction 0 1
# 0 35 20
# 1 20 26
# Accuracy : 0.604
# 95% CI : (0.5017, 0.6999)
# No Information Rate : 0.5446
# P-Value [Acc> NIR] : 0.1357
```

```
#
# Kappa : 0.2016
# Mcnemar's Test P-Value : 1.0000
#
# Sensitivity : 0.6364
# Specificity : 0.5652
# PosPredValue : 0.6364
# NegPredValue : 0.5652
# Prevalence : 0.5446
# Detection Rate : 0.3465
# Detection Prevalence : 0.5446
# Balanced Accuracy : 0.6008
#
# 'Positive' Class : 0
```

CONCLUSION: Thus we have learnt how to Perform the different Data Cleaning and Data modeling operations using Python **.**

(All students are expected to take printout of the code which has been executed. It is only for reference)

EXPERIMENT 6

TITLE: Integrate Python and Hadoop and perform the following operations on forest fire dataset

OBJECTIVE:

1. To understand and apply the Analytical concept of Big data using Python.
2. To study detailed concept RHadoop .

SOFTWARE REQUIREMENTS:

1. Ubuntu 14.04 / 14.10
2. GNU C Compiler
3. Hadoop
4. Java

PROBLEM STATEMENT: Integrate Python and Hadoop and perform the following operations on forest fire dataset

1) Data analysis using the Map Reduce in PyHadoop

2) Data mining in Hive

THEORY:

Write theory by your own for

1. **PYHadoop** can be installed using command - **pip install pyhadoop**
   Pydoop is a Python interface to Hadoop that allows you to write MapReduce applications in pure Python. Pydoop Script is the easiest way to write simple MapReduce programs for Hadoop. With Pydoop Script, you only need to write a map and/or a reduce functions and the system will take care of the rest.

Command Line Tool

In the simplest case, Pydoop Script is invoked as:

```
pydoop script MODULE INPUT OUTPUT
```

where MODULE is the file (on your local file system) containing your map and reduce functions, in Python, while INPUT and OUTPUT are, respectively, the HDFS paths of your input data and your job's output directory.

Options are shown in the following table.

Short Long Meaning

--num-reducers Number of reduce tasks. Specify 0 to only perform map phase

--no-override home

--no-override ld-path

--no-override pypath

--no-override env

--no-override path
Don't set the script's HOME directory to the $HOME in your environment. Hadoop will set it to the value of the 'mapreduce.admin.user.home.dir' property
Use the default PATH, LD_LIBRARY_PATH and PYTHONPATH, instead of copying them from the submitting client node

Use the default LD_LIBRARY_PATH instead of copying it from the submitting client node
Use the default PYTHONPATH instead of copying it from the submitting client node
Use the default PATH instead of copying it from the submitting client node

--set-env Set environment variables for the tasks. If a variable is set to '', it will not be overridden by Pydoop.

-D --job-conf Set a Hadoop property, e.g., -D mapreduce.job.priority=high

--python-zip Additional python zip file

--upload-file to-cache Upload and add this file to the distributed cache.

--upload archive-to cache Upload and add this archive file to the distributed cache.

--log-level Logging level

--job-name name of the job

--python program python executable that should be used by the wrapper

--pretend Do not actually submit a job, print the generated config settings and the command line that would be invoked

--hadoop-conf Hadoop configuration file

--input-format java classname of InputFormat

-m --map-fn name of map function within module

-r --reduce-fn name of reduce function within module

-c --combine-fn name of combine function within module

--combiner-fn –combine-fn alias for backwards compatibility

-t --kv-separator output key-value separator

For more Information Refer - https://crs4.github.io/pydoop/pydoop_script.html

2. **Data Mining in Hive** - Hive is a batch-oriented and data -warehousing layer created on the basic elements of Hadoop , such as HDFS and mapreduce. This layer plays an important role in mining of big data . Hive offers a simple SQL -lite-implementation call hiveQL to SQL users without losing access through mappers and reducers.

Apache Hive is a data warehouse and an ETL tool which provides an SQL-like interface between

the user and the Hadoop distributed file system (HDFS) which integrates Hadoop. It is built on top of Hadoop. It is a software project that provides data query and analysis. It facilitates reading, writing and handling wide datasets that stored in distributed storage and queried by Structure Query Language (SQL) syntax. It is not built for Online Transactional Processing (OLTP) workloads. It is frequently used for data warehousing tasks like data encapsulation, Ad-hoc Queries, and analysis of huge datasets. It is designed to enhance scalability, extensibility, performance, fault-tolerance and loose-coupling with its input formats.

Initially Hive is developed by Facebook and Amazon, Netflix and It delivers standard SQL functionality for analytics. Traditional SQL queries are written in the MapReduce Java API to execute SQL Application and SQL queries over distributed data. Hive provides portability as most data warehousing applications functions with SQL-based query languages like NoSQL.

Components of Hive:

HCatalog –

It is a Hive component and is a table as well as a store management layer for Hadoop. It enables user along with various data processing tools like Pig and MapReduce which enables to read and write on the grid easily.

WebHCat –

It provides a service which can be utilized by the user to run Hadoop MapReduce (or YARN), Pig, Hive tasks or function Hive metadata operations with an HTTP interface.

Modes of Hive:

Local Mode –

It is used, when the Hadoop is built under pseudo mode which has only one data node, when the data size is smaller in term of restricted to single local machine, and when processing will be faster on smaller datasets existing in the local machine.

Map Reduce Mode –

It is used, when Hadoop is built with multiple data nodes and data is divided across various nodes, it will function on huge datasets and query is executed parallelly, and to achieve enhanced performance in processing large datasets.

EXPERIMENT 7

TITLE

Visualize the data using Python libraries matplotlib, seaborn by plotting the graphs for assignment no. 2 and 3 (Group B)

OBJECTIVE:

1.To understand and apply the Analytical concept of Big data using

Python. 2. To study detailed concept Python.

SOFTWARE REQUIREMENTS:

1. Ubuntu 14.04 / 14.10
2. GNU C Compiler
3. Hadoop
4. Java

PROBLEM STATEMENT: Visualize the data using Python libraries matplotlib, seaborn by plotting the graphs for assignment no. 2 and 3 (Group B)

THEORY:

Data Visualisation in Python using Matplotlib and Seaborn

Data visualization is an easier way of presenting the data, however complex it is, to analyze trends and relationships amongst variables with the help of pictorial representation.

The following are the advantages of Data Visualization

- Easier representation of compels data
- Highlights good and bad performing areas

• Explores relationship between data points
• Identifies data patterns even for larger data points

While building visualization, it is always a good practice to keep some below mentioned points in mind

• Ensure appropriate usage of shapes, colors, and size while building visualization • Plots/graphs using a co-ordinate system are more pronounced
• Knowledge of suitable plot with respect to the data types brings more clarity to the information • Usage of labels, titles, legends and pointers passes seamless information the wider audience

Python Libraries

There are a lot of python libraries which could be used to build visualization like *matplotlib, vispy, bokeh, seaborn, pygal, folium, plotly, cufflinks*, and *networkx*. Of the many, *matplotlib* and *seaborn* seems to be very widely used for basic to intermediate level of visualizations.

Matplotlib

It is an amazing visualization library in Python for 2D plots of arrays, It is a multi-platform data visualization library built on *NumPy* arrays and designed to work with the broader *SciPy* stack. It was introduced by John Hunter in the year 2002. Let's try to understand some of the benefits and features of *matplotlib*

• It's fast, efficient as it is based on *numpy* and also easier to build
• Has undergone a lot of improvements from the open source community since inception and hence a better library having advanced features as well
• Well maintained visualization output with high quality graphics draws a lot of users to it • Basic as well as advanced charts could be very easily built
• From the users/developers point of view, since it has a large community support, resolving issues and debugging becomes much easier

Seaborn

Conceptualized and built originally at the Stanford University, this library sits on top of *matplotlib*. In a sense, it has some flavors of *matplotlib* while from the visualization point, its is much better than *matplotlib* and has added features as well. Below are its advantages

• Built-in themes aid better visualization
• Statistical functions aiding better data insights
• Better aesthetics and built-in plots
• Helpful documentation with effective examples

Nature of Visualization

Depending on the number of variables used for plotting the visualization and the type of variables, there could be different types of charts which we could use to understand the relationship. Based on the count of variables, we could have

• *Univariate* plot(involves only one variable)
• *Bivariate* plot(more than one variable in required)

A *Univariate* plot could be for a continuous variable to understand the spread and distribution of the variable while for a discrete variable it could tell us the count

Similarly, a *Bivariate* plot for continuous variable could display essential statistic like correlation, for a continuous versus discrete variable could lead us to very important conclusions like understanding data distribution across different levels of a categorical variable. A *bivariate* plot between two discrete variables could also be developed.

Box plot

A boxplot, also known as a box and whisker plot, the box and the whisker are clearly displayed in the below image. It is a very good visual representation when it comes to measuring the data distribution. Clearly plots the median values, outliers and the quartiles. Understanding data distribution is another important factor which leads to better model building. If data has outliers, box plot is a recommended way to identify them and take necessary actions.

Syntax: *seaborn.boxplot(x=None, y=None, hue=None, data=None, order=None, hue_order=None, orient=None, color=None, palette=None, saturation=0.75, width=0.8, dodge=True, fliersize=5, linewidth=None, whis=1.5, ax=None, **kwargs)*

Parameters:

x, y, hue: *Inputs for plotting long-form data.*

data: *Dataset for plotting. If x and y are absent, this is interpreted as wide-form.*

color: *Color for all of the elements.*

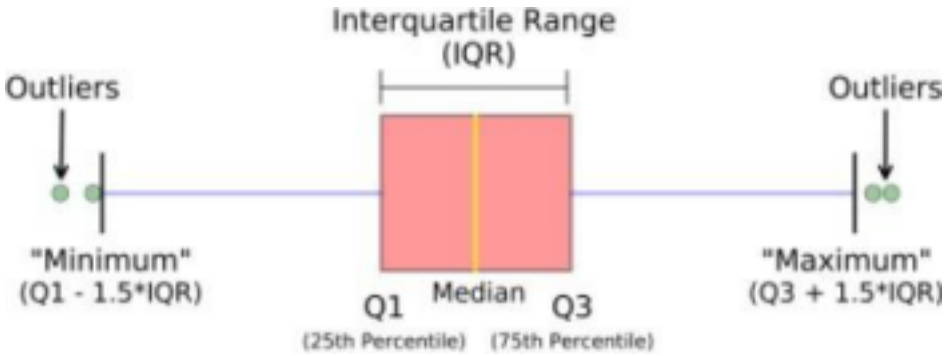
Returns: *It returns the Axes object with the plot drawn onto it.*

The box and whiskers chart shows how data is spread out. Five pieces of information are generally included in the chart

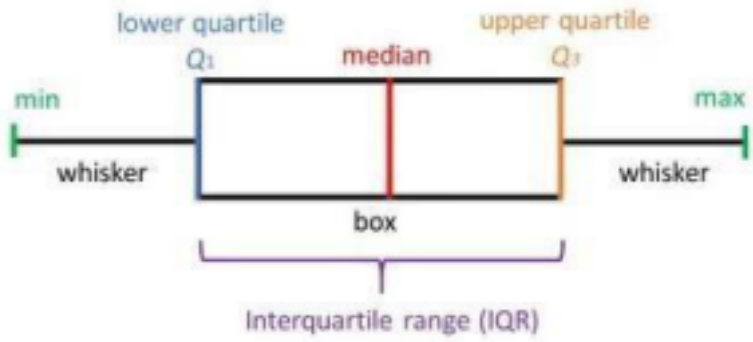
1. The minimum is shown at the far left of the chart, at the end of the left 'whisker'
2. First quartile, Q1, is the far left of the box (left whisker)
3. is shown as a line in the center of the box The median
4. Third quartile, Q3, shown at the far right of the box (right whisker)
5. The maximum is at the far right of the box

As could be seen in the below representations and charts, a box plot could be plotted for one or more than one variable providing very good insights to our data.

Representation of box plot.



Box plot representing multi-variate categorical variables



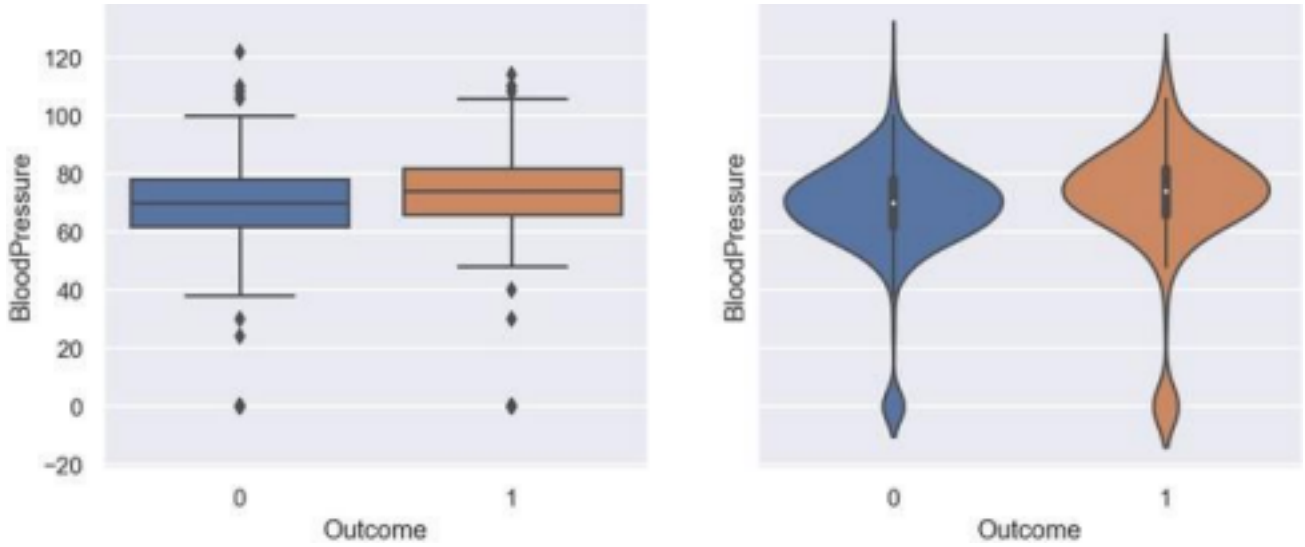
Box plot representing multi-variate categorical variables

• Python3

```
# import required modules

import matplotlib as plt

import seaborn as sns



# Box plot and violin plot for Outcome vs BloodPressure
```



```
_, axes = plt.subplots(1, 2, sharey=True, figsize=(10, 4))



# box plot illustration

sns.boxplot(x='Outcome', y='BloodPressure', data=diabetes,
ax=axes[0])



# violin plot illustration

sns.violinplot(x='Outcome', y='BloodPressure', data=diabetes,
ax=axes[1])
```



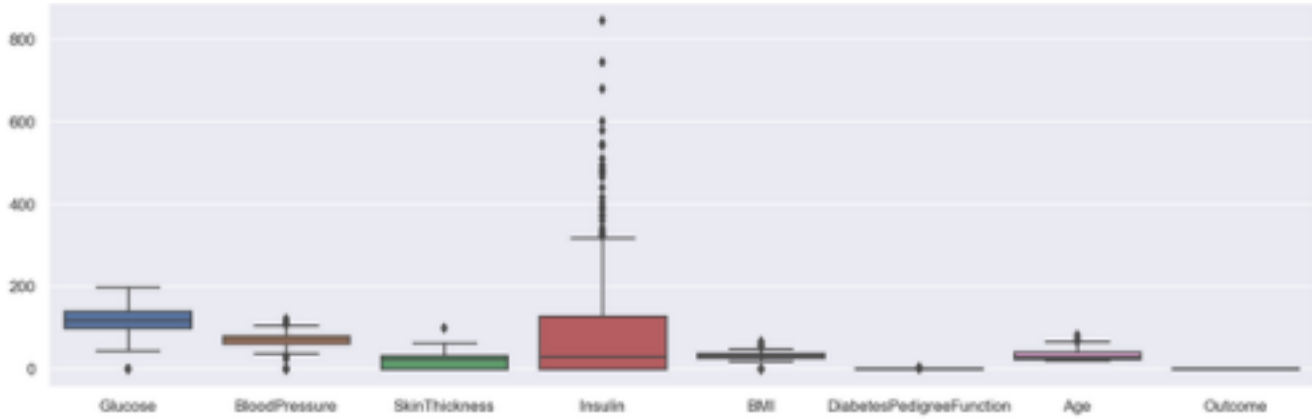
Output for Box Plot and Violin Plot

· Python3

```
# Box plot for all the numerical variables

sns.set(rc={'figure.figsize': (16, 5)})
```



```
# multiple box plot illustration

sns.boxplot(data=diabetes.select_dtypes(include='number'))
```



Output Multiple Box PLot

Scatter Plot

Scatter plots or scatter graphs is a *bivariate* plot having greater resemblance to line graphs in the way they are built. A line graph uses a line on an X-Y axis to plot a continuous function, while a scatter plot relies on dots to represent individual piecesof data. These plots are very useful to see if two variables are correlated. Scatter plotcould be 2 dimensional or 3 dimensional. ***Syntax:*** *seaborn.scatterplot(x=None, y=None, hue=None, style=None, size=None,data=None, palette=None, hue_order=None, hue_norm=None, sizes=None, size_order=None, size_norm=None, markers=True, style_order=None, x_bins=None, y_bins=None, units=None, estimator=None, ci=95, n_boot=1000, alpha='auto', x_jitter=None, y_jitter=None, legend='brief', ax=None, **kwargs)* ***Parameters:***

x, y: *Input data variables that should be numeric.*

data*: Dataframe where each column is a variable and each row is an observation.*

size*: Grouping variable that will produce points with different sizes.*

style*: Grouping variable that will produce points with different markers.* ***palette****: Grouping variable that will produce points with different markers.****markers****: Object determining how to draw the markers for different levels.****alpha****: Proportional opacity of the points.*

Returns: *This method returns the Axes object with the plot drawn onto it.*

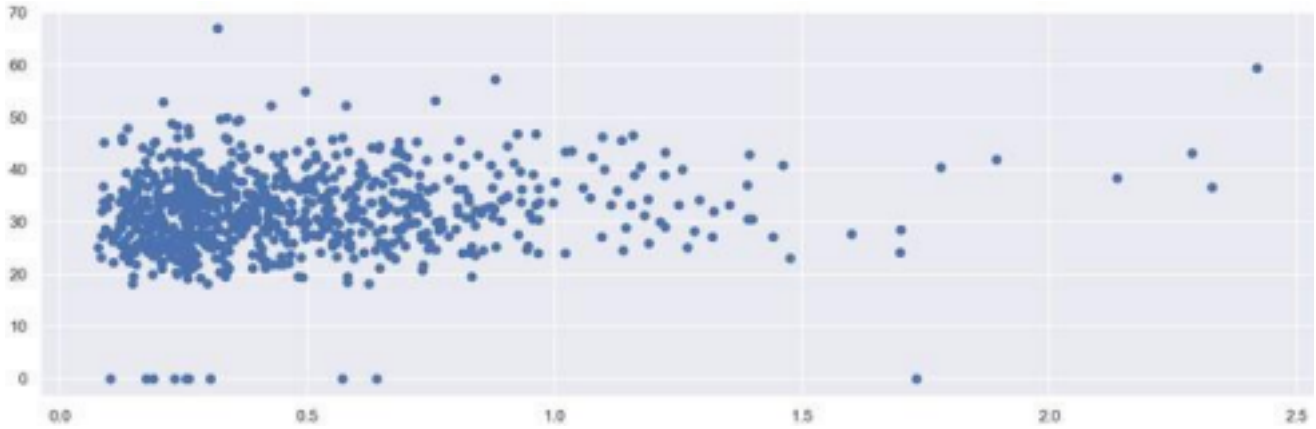
Advantages of a scatter plot

- Displays correlation between variables
- Suitable for large data sets
- Easier to find data clusters
- Better representation of each data point

- Python3

```
# import module

import matplotlib.pyplot as plt



# scatter plot illustration
```

```
plt.scatter(diabetes['DiabetesPedigreeFunction'], diabetes['BMI'])
```



Output 2D Scattered Plot

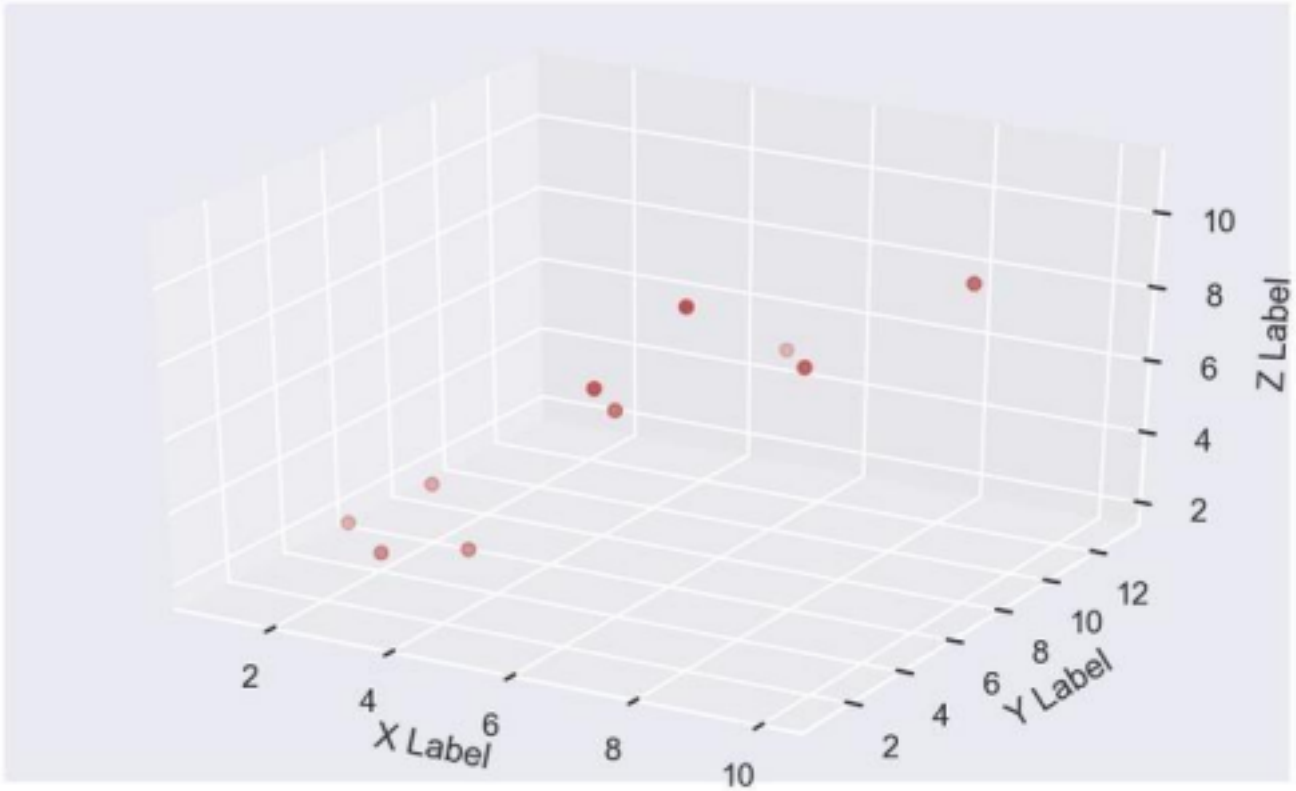
- Python3

```
# import required modules

from mpl_toolkits.mplot3d import Axes3D
```



```
# assign axis values

x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

y = [5, 6, 2, 3, 13, 4, 1, 2, 4, 8]

z = [2, 3, 3, 3, 5, 7, 9, 11, 9, 10]



# adjust size of plot

sns.set(rc={'figure.figsize': (8, 5)})

fig = plt.figure()
```

```
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x, y, z, c='r', marker='o')



# assign labels

ax.set_xlabel('X Label'), ax.set_ylabel('Y Label'), ax.set_zlabel('Z Label')



# display illustration

plt.show()
```



Output 3D Scattered Plot

Histogram

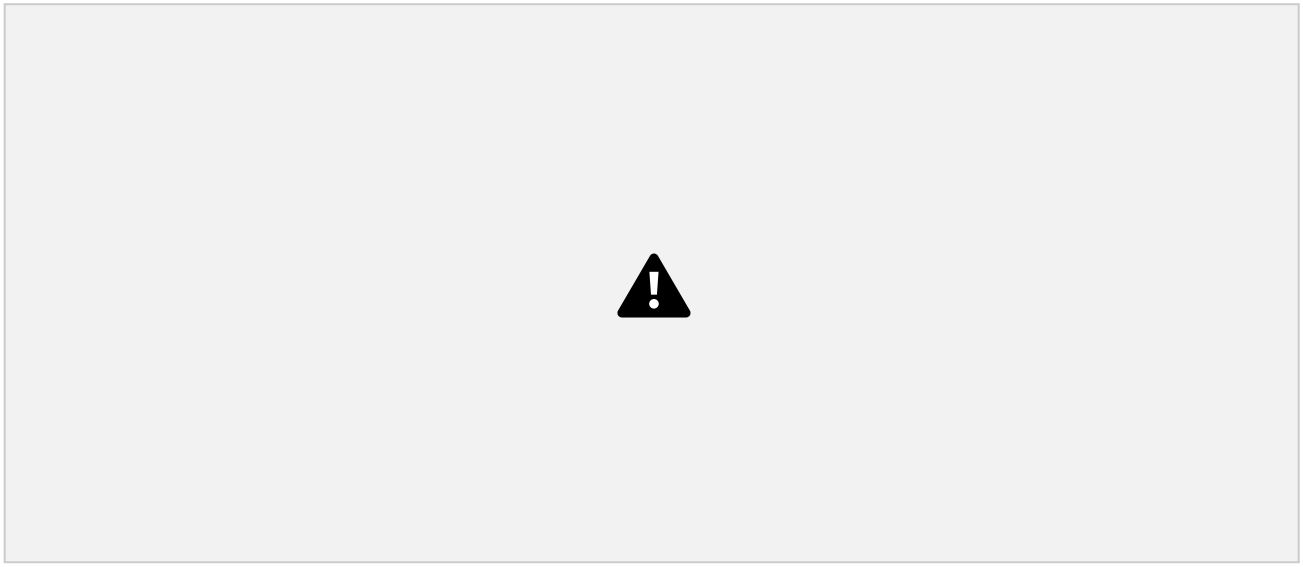
Histograms display counts of data and are hence similar to a bar chart. A histogram plot can also tell us how close a data distribution is to a normal curve. While workingout statistical method, it is very important that we have a data which is normally or close to a normal distribution. However, histograms are *univariate* in nature and bar charts *bivariate*.

A bar graph charts actual counts against categories e.g. height of the bar indicates thenumber of items in that category whereas a histogram displays the same categorical variables in *bins*. Bins are integral part while building a histogram they control the data points which are within a range. As a widely accepted choice we usually limit bin to a size of 5-20,however this is totally governed by the data points which is present.

- Python3

```
# illustrate histogram

features = ['BloodPressure', 'SkinThickness']
```

```
diabetes[features].hist(figsize=(10, 4))
```



Output Histogram

Pie Chart

Pie chart is a *univariate* analysis and are typically used to show percentage or proportional data. The percentage distribution of each class in a variable is provided next to the corresponding slice of the pie. The python libraries which could be used to build a pie chart is *matplotlib* and *seaborn.*
Syntax: *matplotlib.pyplot.pie(data, explode=None, labels=None, colors=None, autopct=None, shadow=False)*
Parameters:
data *represents the array of data values to be plotted, the fractional area of each slice is represented by* ***data/sum(data)****. If sum(data)<1, then the data values returns the fractional area directly, thus resulting pie will have empty wedge of size 1-sum(data).*
labels *is a list of sequence of strings which sets the label of each wedge.*
color *attribute is used to provide color to the wedges.*
autopct *is a string used to label the wedge with their numerical value.*
shadow *is used to create shadow of wedge.*
Below are the advantages of a pie chart

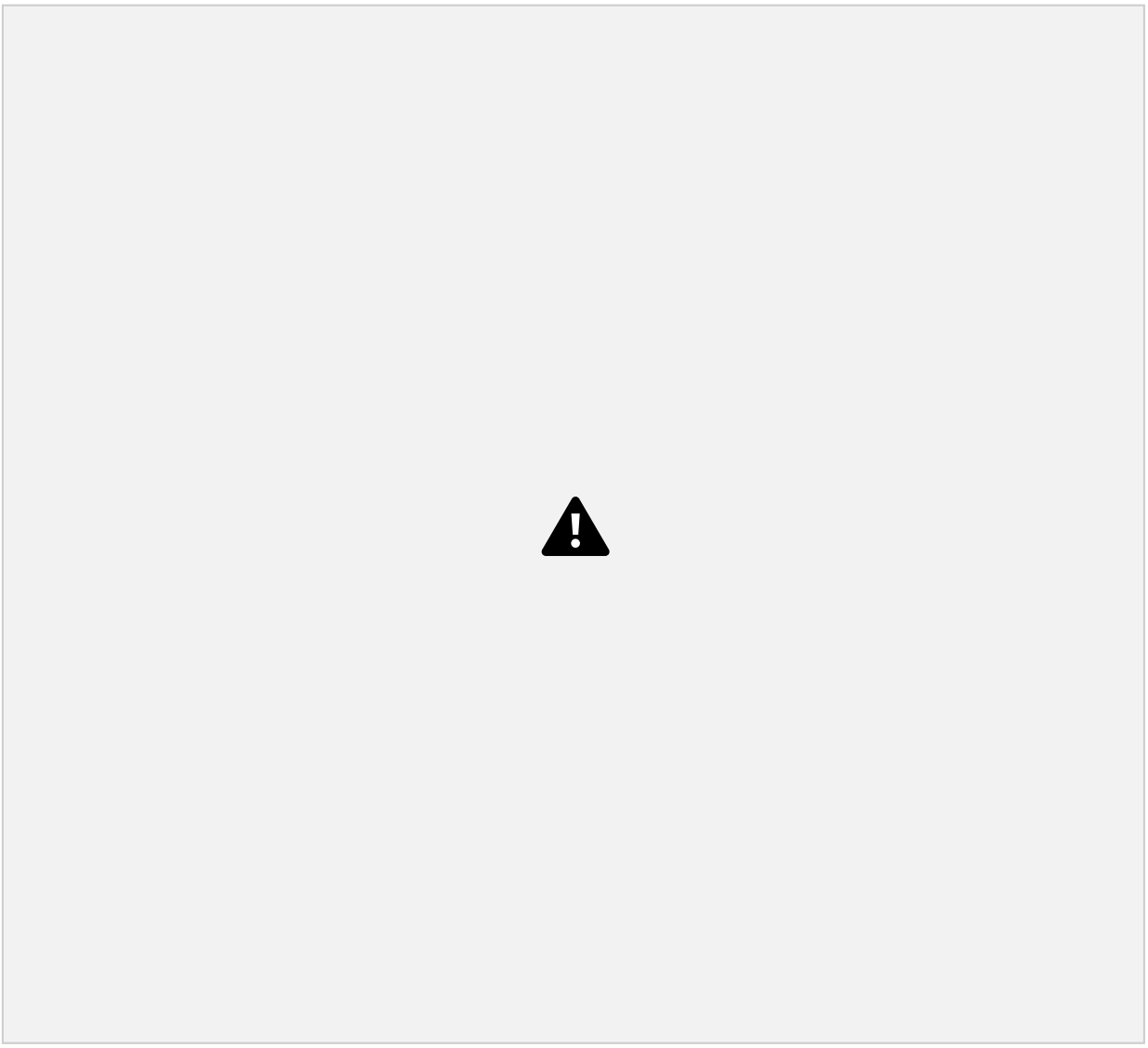
- Easier visual summarization of large data points
- Effect and size of different classes can be easily understood
- Percentage points are used to represent the classes in the data points

- Python3

```
# import required module
```



```
import matplotlib.pyplot as plt


# Creating dataset

cars = ['AUDI', 'BMW', 'FORD', 'TESLA', 'JAGUAR',
'MERCEDES']

data = [23, 17, 35, 29, 12, 41]


# Creating plot

fig = plt.figure(figsize=(10, 7))

plt.pie(data, labels=cars)


# Show plot

plt.show()
```



Output Pie Chart

- Python3

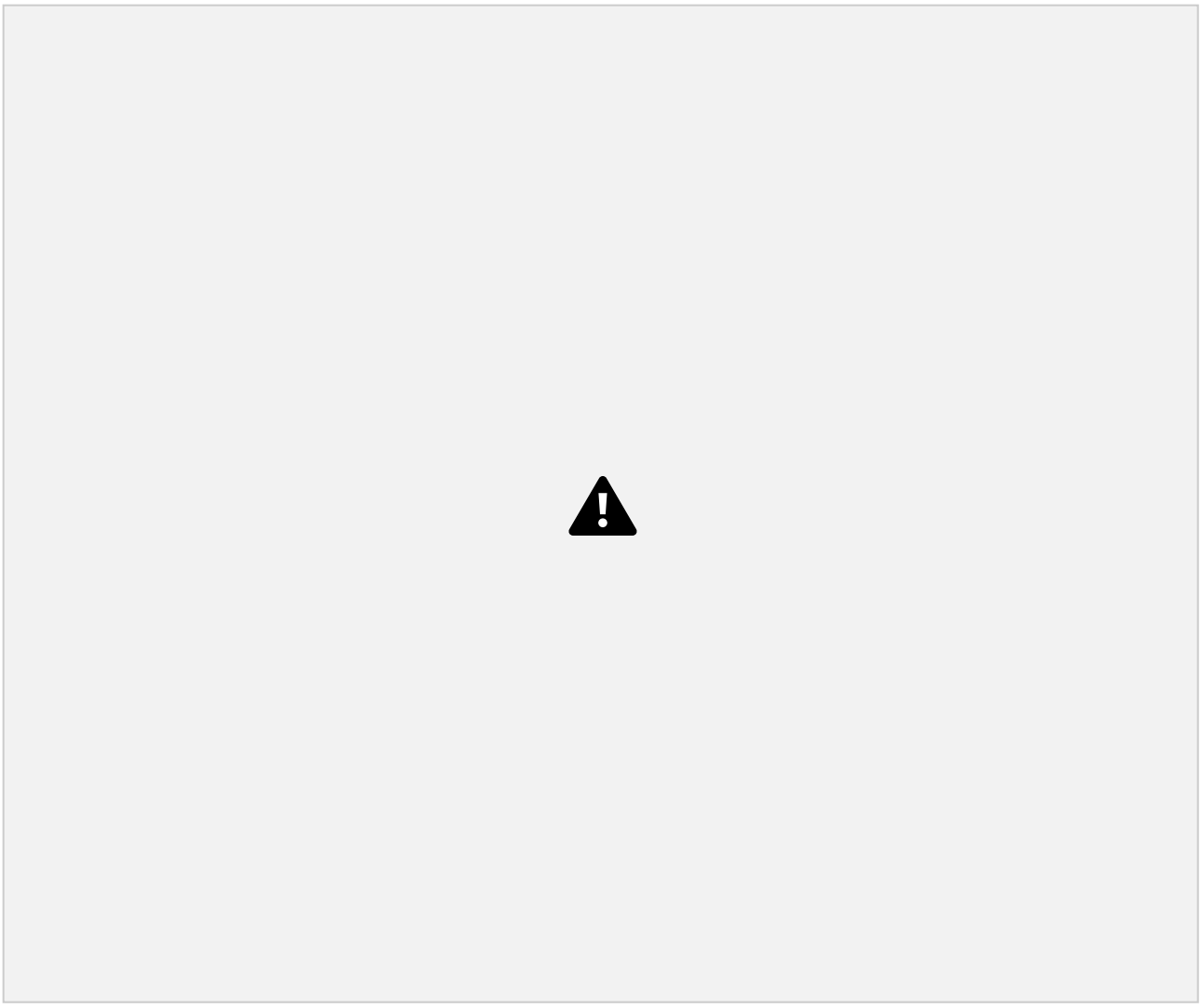
```
# Import required module

import matplotlib.pyplot as plt

import numpy as np
```

```
# Creating dataset

cars = ['AUDI', 'BMW', 'FORD', 'TESLA', 'JAGUAR',
'MERCEDES']

data = [23, 17, 35, 29, 12, 41]



# Creating explode data

explode = (0.1, 0.0, 0.2, 0.3, 0.0, 0.0)



# Creating color parameters

colors = ("orange", "cyan", "brown", "grey", "indigo", "beige")



# Wedge properties

wp = {'linewidth': 1, 'edgecolor': "green"}



# Creating autocpt arguments

def func(pct, allvalues):

    absolute = int(pct / 100.*np.sum(allvalues))

    return "{:.1f}%\n({:d} g)".format(pct, absolute)
```



```
# Creating plot
```

```
fig, ax = plt.subplots(figsize=(10, 7))

wedges, texts, autotexts = ax.pie(data, autopct=lambda
pct: func(pct, data), explode=explode, labels=cars,

                          shadow=True, colors=colors, startangle=90,
wedgeprops=wp,

                          textprops=dict(color="magenta"))



# Adding legend

    ax.legend(wedges, cars, title="Cars", loc="center

            left", bbox_to_anchor=(1, 0, 0.5, 1))

plt.setp(autotexts, size=8, weight="bold")

ax.set_title("Customizing pie chart")



# Show plot

plt.show()
```



Output

CONCLUSION: Thus we have learnt Visualize the data using Python by plotting the graphs .

ASSIGNMENT-8

Aim:

Perform the following data visualization operations using Tableau on Adult and Iris datasets

1) 1D (Linear) Data visualization

2) 2D (Planar) Data Visualization

3) 3D (Volumetric) Data Visualization

4) Temporal Data Visualization

5) Multidimensional Data Visualization

6) Tree/ Hierarchical Data visualization

7) Network Data visualization

- Introduction

Data visualization or data visualization is viewed by many disciplines as a modern equivalent of visual communication. It involves the creation and study of the visual representation of data, meaning "information that has been abstracted in some schematic form, including attributes or variables for the units of information".

Data visualization refers to the techniques used to communicate data or information by encoding it as visual objects (e.g., points, lines or bars) contained in graphics. The goal is to communicate information clearly and efficiently to users. It is one of the steps in data analysis or data science

1D/Linear

Examples:

- lists of data items, organized by a single feature (e.g., alphabetical order)
  (not commonly visualized)

2D/Planar (especially geospatial)

Examples (geospatial):

- **choropleth**

3D/Volumetric

3D/Volumetric

Broadly, examples of scientific visualization:

- 3D computer models

In 3D computer graphics, **3D modeling** (or **three-dimensional modeling**) is the process of developing a mathematical representation of any *surface* of an object (either inanimate or living) in three dimensions via specialized software. The product is called a **3D model**. Someone who works with 3D models may be referred to as a **3D artist**. It can be displayed as a two-dimensional image through a process called *3D rendering* or used in a computer simulation of physical phenomena. The model can also be physically created using 3D printing devices.

- surface and volume rendering

Rendering is the process of generating an image from a model, by means of computer programs. The model is a description of three-dimensional objects in a strictly defined language or data structure. It would contain geometry, viewpoint, texture, lighting, and shading information. The image is a digital image or raster graphics image. The term may be by analogy with an "artist's rendering" of a scene. 'Rendering' is also used to describe the process of calculating effects in a video editing file to produce final video output.

Volume rendering is a technique used to display a 2D projection of a 3D discretely sampled data set. A typical 3D data set is a group of 2D slice images acquired by a CT or MRI scanner. Usually these are acquired in a regular pattern (e.g., one slice every millimeter) and usually have a regular number of image pixels in a regular pattern. This is an example of a regular volumetric grid, with

each volume element, or voxel represented by a single value that is obtained by sampling the immediate area surrounding the voxel.

· computer simulations

Computer simulation is a computer program, or network of computers, that attempts to simulate an abstract model of a particular system. Computer simulations have become a useful part of mathematical modeling of many natural systems in physics, and computational physics, chemistry and biology; human systems in economics, psychology, and socialscience; and in the process of engineering and new technology, to gain insight into the operation of those systems, or to observe their behavior.[6] The simultaneous visualization and simulation of a system is called visulation.

Temporal

Examples:

· **timeline**

Tools: SIMILE Timeline, TimeFlow, Timeline JS, Excel

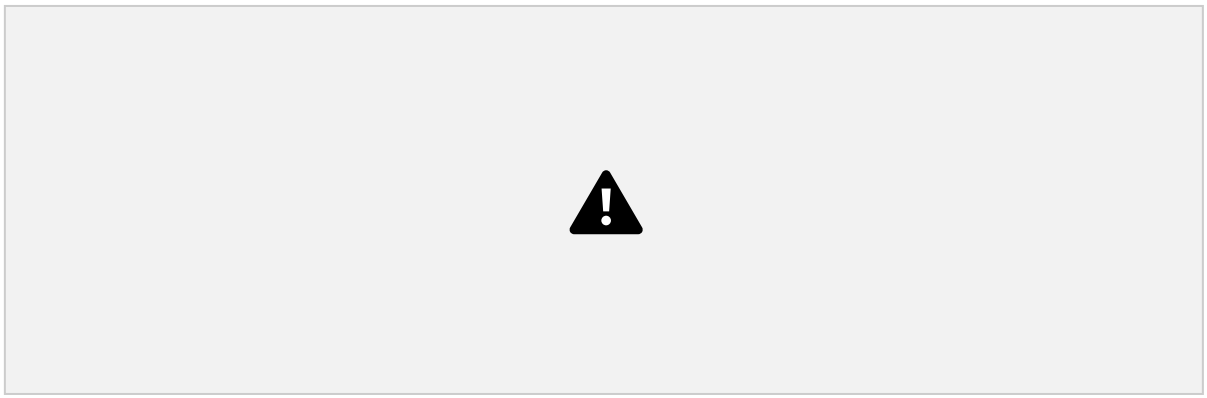
Image:

Friendly, M. & Denis, D. J. (2001). Milestones in the history of thematic cartography,

statistical graphics, and data visualization.Web

document, http://www.datavis.ca/milestones/.Accessed: August 30, 2012.

- **time series**

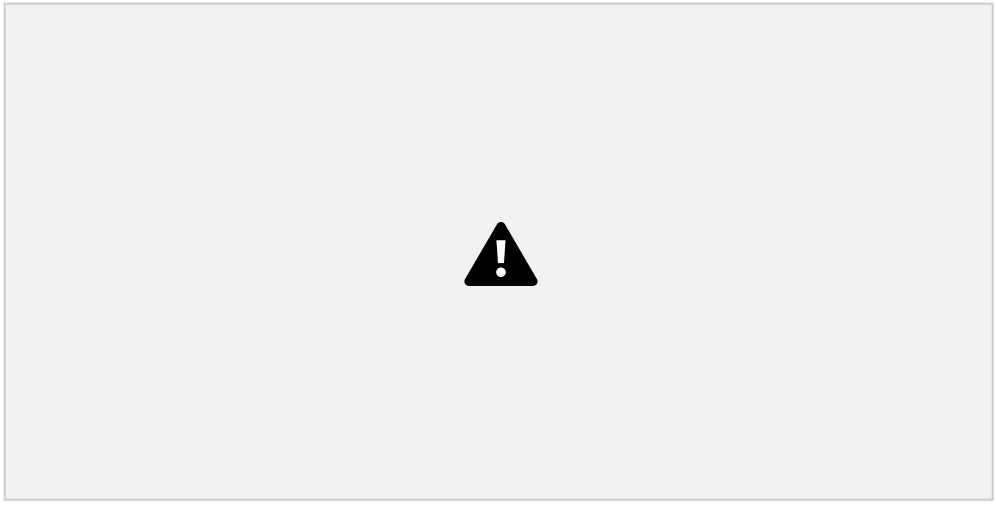


nD/Multidimensional

Examples (category proportions, counts):

- **histogram**

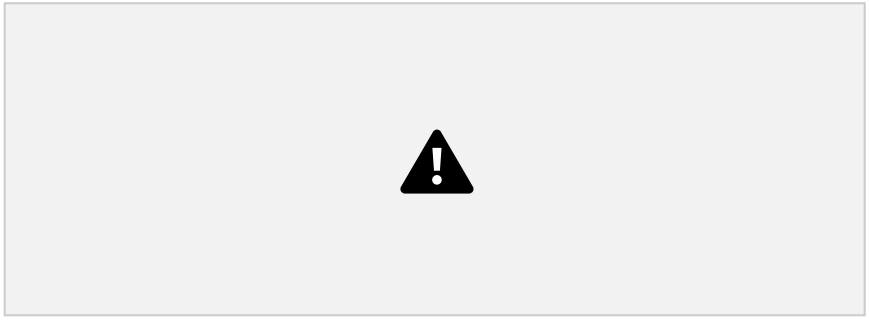
- **pie chart**



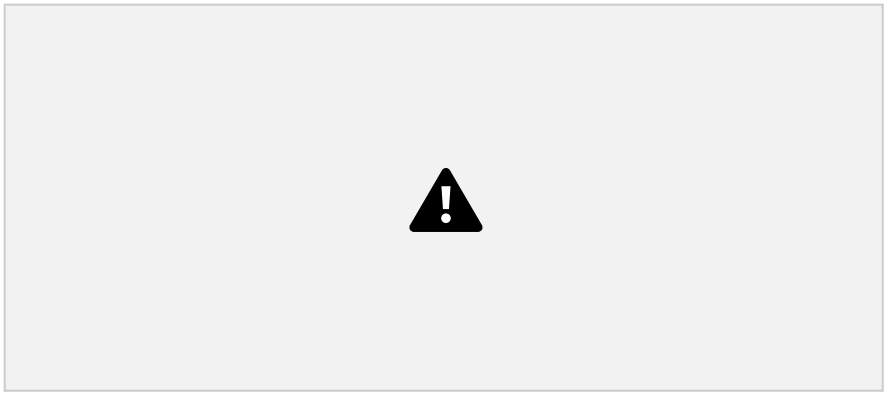
Tree/Hierarchical

Examples:

- **general tree visualization**



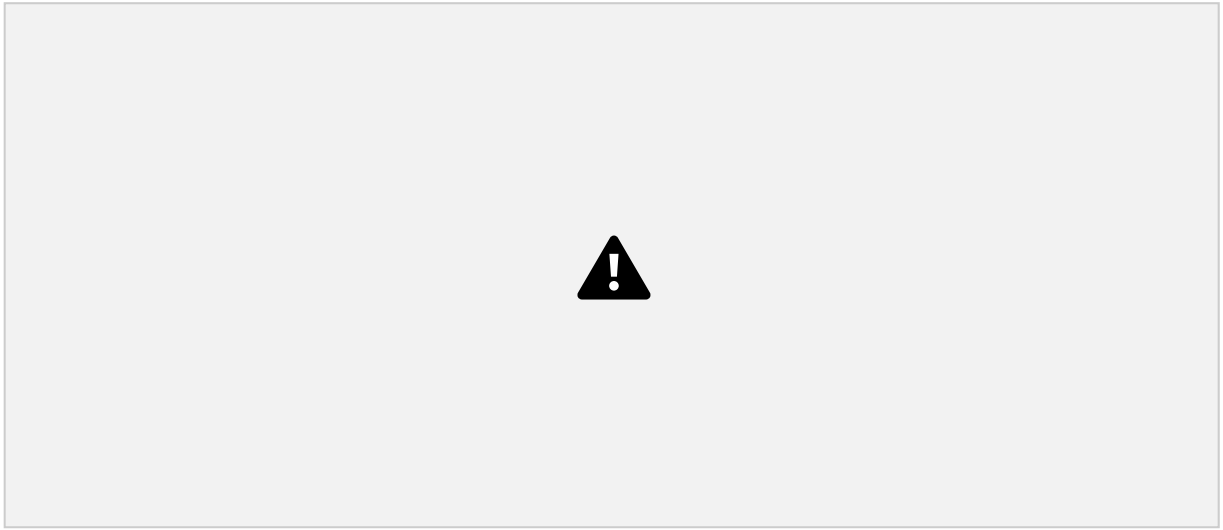
- **dendrogram**



Network

~~Examples:~~

- **matrix**



- **node-link diagram (link-based layout algorithm)**

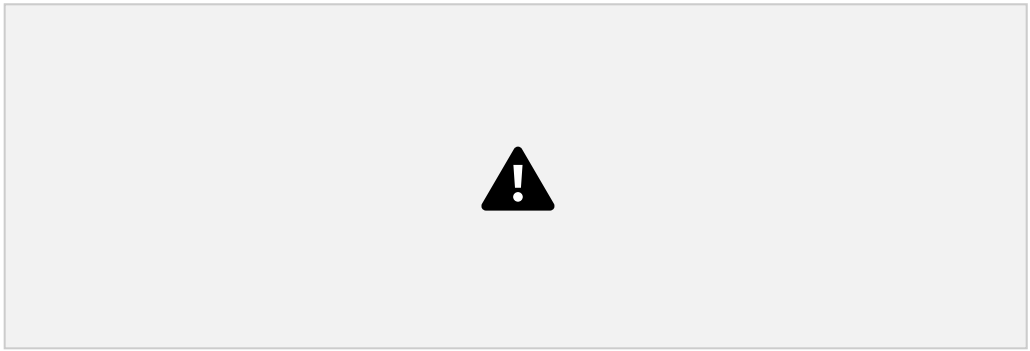


Tableau:

Tableau is a Business Intelligence tool for visually analyzing the data. Users can create and distribute an interactive and shareable dashboard, which depict the trends, variations, and density of

the data in the form of graphs and charts. Tableau can connect to files, relational and Big Data sources to acquire and process data. The software allows data blending and real-time collaboration, which makes it very unique. It is used by businesses, academic researchers, and many government organizations for visual

data analysis. It is also positioned as a leader Business Intelligence and Analytics Platform in Gartner Magic Quadrant.

Tableau Features:

Tableau providessolutions for all kinds of industries, departments, and data environments. Following are some unique features which enable Tableau to handle diverse scenarios.

- **Speed of Analysis –** As it does not require high level of programming expertise, any user with access to data can start using it to derive value from the data.
- **Self-Reliant –** Tableau does not need a complex software setup. The desktop version which is used by most users is easily installed and contains all the features needed to start and complete data analysis.
- **Visual Discovery –** The user explores and analyzes the data by using visual tools like colors, trend lines, charts, and graphs. There is very little script to be written as nearly everything is done by drag and drop.
- **Blend Diverse Data Sets –** Tableau allows you to blend different relational, semi structured and raw data sources in real time, without expensive up-front integration costs. The users don't need to know the details of how data is stored.
- **Architecture Agnostic –** Tableau works in all kinds of devices where data flows. Hence, the user need not worry about specific hardware or software requirements to use Tableau. ·
Real-Time Collaboration – Tableau can filter, sort, and discuss data on the fly and embed a live dashboard in portals like SharePoint site or Salesforce. You can save your view of data and allow colleagues to subscribe to your interactive dashboards so they see the very latest data just by refreshing their web browser.
- **Centralized Data –** Tableau server provides a centralized location to manage all of the organization's published data sources. You can delete, change permissions, add tags, and manage schedules in one convenient location. It's easy to schedule extract refreshes and

manage them in the data server. Administrators can centrally define a schedule for extracts on the server for both incremental and full refreshes.

There are three basic steps involved in creating any Tableau data analysis report.

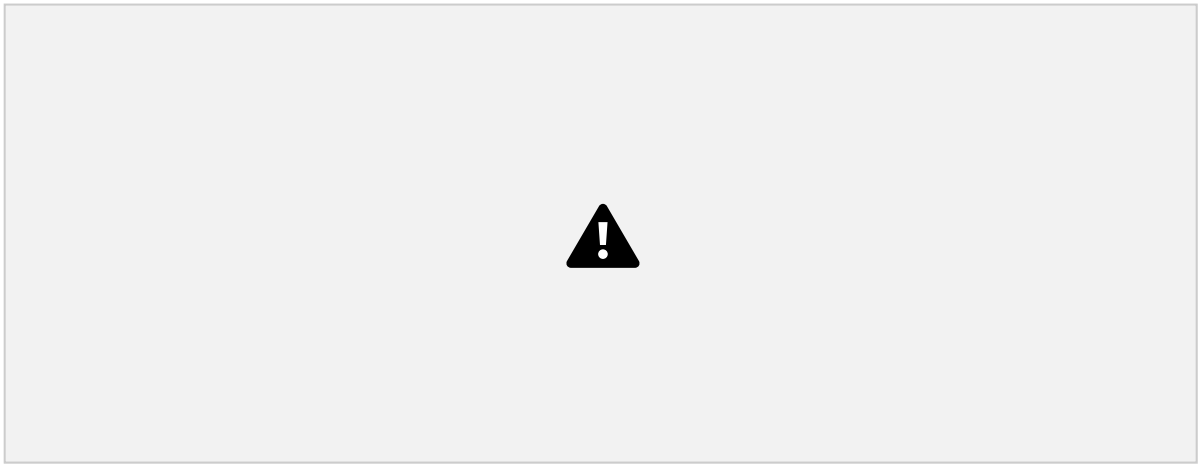
These three steps are –

- **Connect to a data source** – It involves locating the data and using an appropriate type of connection to read the data.
- **Choose dimensions and measures** – This involves selecting the required columns from the source data for analysis.
- **Apply visualization technique** – This involves applying required visualization methods, such as a specific chart or graph type to the data being analyzed.

For convenience, let's use the sample data set that comes with Tableau installation named sample – superstore.xls. Locate the installation folder of Tableau and go to **My Tableau Repository**. Under it, you will find the above file at **Datasources\9.2\en_US-US**.

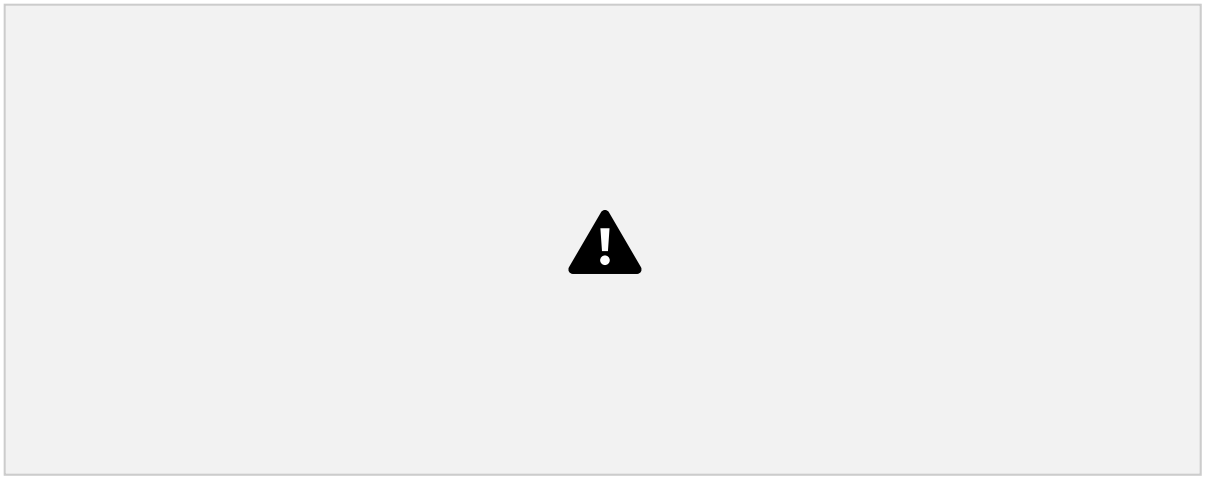
Connect to a Data Source

On opening Tableau, you will get the start page showing various data sources. Under the header **"Connect"**, you have options to choose a file or server or saved data source. Under Files, choose excel. Then navigate to the file **"Sample – Superstore.xls"** as mentioned above. The excel file has three sheets named Orders, People and Returns. Choose **Orders**.



Choose the Dimensions and Measures

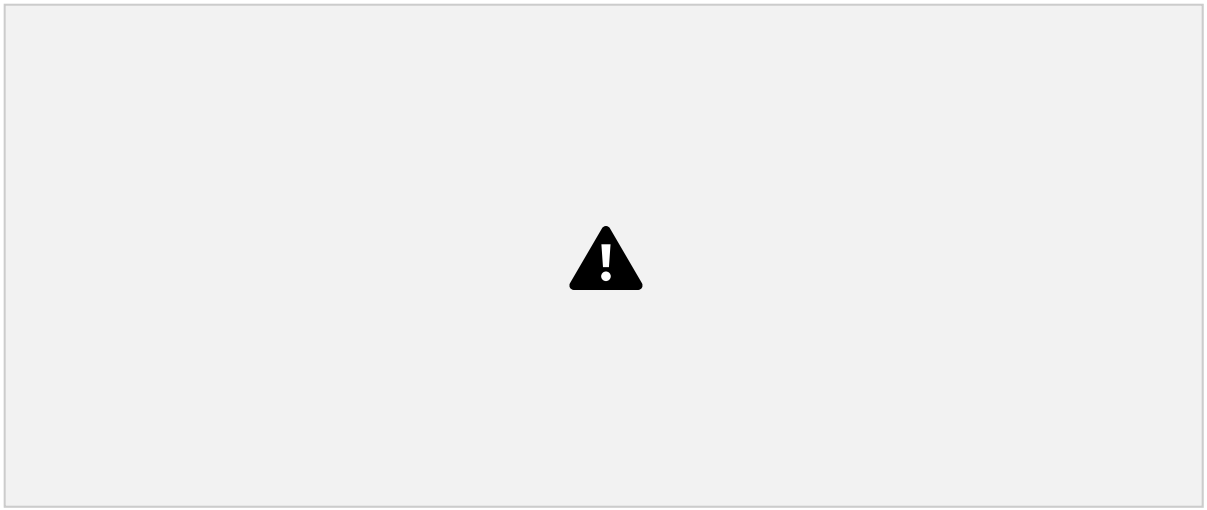
Next, choose the data to be analyzed by deciding on the dimensions and measures. Dimensions are the descriptive data while measures are numeric data. When put together, they help visualize the performance of the dimensional data with respect to the data which are measures. Choose **Category** and **Region** as the dimensions and **Sales** as the measure. Drag and drop them as shown in the following screenshot. The result shows the total sales in each category for each region.



Apply Visualization Technique

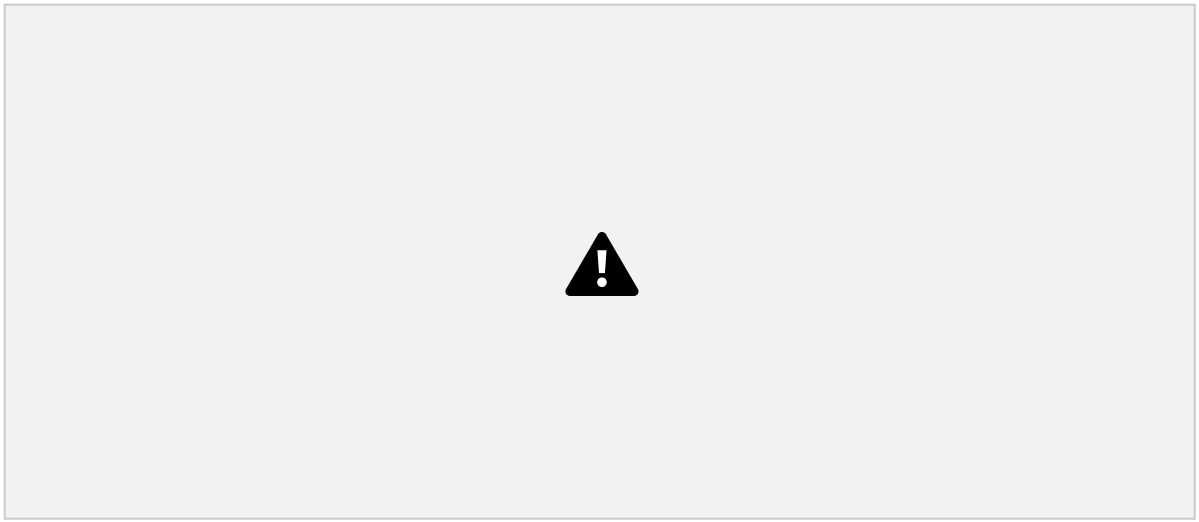
In the previous step, you can see that the data is available only as numbers. You have to read and calculate each of the values to judge the performance. However, you can see them as graphs or charts with different colors to make a quicker judgment.

We drag and drop the sum (sales) column from the Marks tab to the Columns shelf. The table showing the numeric values of sales now turns into a bar chart automatically.



You can apply a technique of adding another dimension to the existing data. This will add

more colors to the existing bar chart as shown in the following screenshot.



Conclusion: Thus we have learnt how to Visualize the data in different types (1 1D (Linear) Data visualization,2D (Planar) Data Visualization, 3D (Volumetric) Data Visualization, Temporal Data Visualization, Multidimensional Data Visualization, Tree/ Hierarchical Data visualization, Network Data visualization) by using Tableau Software.

Group C: Model Implementation

Aim – To create a Web Scraper

Title -

1.Create a review scrapper for any ecommerce website to fetch real time comments, reviews, ratings,comment tags, customer name using Python.

Introduction - **Web scraping**, **web harvesting**, or **web data extraction** is data scraping used for extracting data from websites. The web scraping software may directly access the World Wide Web using the Hypertext Transfer Protocol or a web browser. While web scraping can be done manually by a software user, the term typically refers to automated processes implemented using a bot or web crawler. It is a form of copying in which specific data is gathered and copied from the web, typically into a central local database or spreadsheet, for later retrieval or analysis.

Web scraping a web page involves fetching it and extracting from it. Fetching is the downloading of a page (which a browser does when a user views a page). Therefore, web crawling is a main component of web scraping, to fetch pages for later processing. Once fetched, then extraction can take place. The content of a page may be parsed, searched, reformatted, its data copied into a spreadsheet or loaded into a database. Web scrapers typically take something out of a page, to make use of it for another purpose somewhere else. An example would be to find and copy names and telephone numbers, or companies and their URLs, or e-mail addresses to a list (contact scraping).

Beautiful Soup is a Python package for parsing HTML and XML documents (including having malformed markup, i.e. non-closed tags, so named after tag soup). It creates a parse tree for parsed pages that can be used to extract data from HTML,[3] which is useful for web scraping.

Beautiful Soup was started by Leonard Richardson, who continues to contribute to the project, and is additionally supported by Tidelift, a paid subscription to open-source maintenance.

It is available for Python 2.7 and Python 3.

CODE –

```
!pip install BeautifulSoup4
!pip install requests
from bs4 import BeautifulSoup as bs
import requests
link =
'https://meesho.com/classic-fancy-bedsheets/p/f6j5p?view_mode=all_r_n_r' page
= requests.get(link)
page
page.content
soup = bs(page.content,'html.parser')
print(soup.prettify())
names = soup.find_all('span',class_='Text__StyledText-sc-oo0kvp-0
cwTMA-d') names
cust_name = []
for i in range(0,len(names)):
```



```
  cust_name.append(names[i].get_text())
cust_name
output ['Sunyana Sharma', 'Usha Marshal Rodrigues']

PRINT PUBLISH DATE

publish_date = soup.find_all('span',class_='Text__StyledText-sc-oo0kvp-0
fMjoAc') publish_date
cust_publish_date = []
for i in range(0,len(publish_date)):
  cust_publish_date.append(publish_date[i].get_text())
cust_publish_date
OUTPUT
['5\xa0 Ratings,', '2\xa0 Reviews', 'Posted on 17 Jul 2021', 'Posted on 15 Oct
2021'] cust_publish_date.pop(0)

REVIEWS
reviews = soup.find_all('span',class_='Text__StyledText-sc-oo0kvp-0 gKkBjb
Comment__CommentText-sc-1ju5q0e-3 kFZtes Comment__CommentText-sc-1ju5q0e-3
kFZtes') reviews
cust_review = []
for i in range(0,len(reviews)):
```

```
    cust_review.append(reviews[i].get_text())
cust_review
OUTPUT

['', 'Same is shown in picture']

REVIEW RATE
review_rate = soup.find_all('span',class_='Text__StyledText-sc-oo0kvp-0
gYxLUd') review_rate
cust_review_rate = []
for i in range(0,len(review_rate)):
    cust_review_rate.append(review_rate[i].get_text())
cust_review_rate

OUTPUT
['3.4', '5.0', '4.0']

CREATE A DATAFRAME TO SHOW ALL THE OUTPUT VALUES
cust_name
cust_publish_date
cust_review
cust_review_rate
```

Conclusion – Hence we have scraped the data from e-commerce website.

2. Develop a mini project in a group using different predictive models techniques to solve any real life problem. (Refer link dataset- https://www.kaggle.com/tanmoyie/us-graduate-schools- admission parameters)

(STUDENTS ARE SUPPOSED TO DO ON YOUR OWN AND SHOW THE SCREEN SHOTS OF THEIR OUTPUT)

Reference Books:

1. Big Data, Black Book, DT Editorial services, 2015 edition.
2. Data Analytics with Hadoop, Jenny Kim, Benjamin Bengfort, OReilly Media, Inc. 3. Python for Data Analysis by Wes McKinney published by O' Reilly media, ISBN : 978-1-449- 31979-3.
4. Python Data Science Handbook by Jake VanderPlas https://tanthiamhuat.files.wordpress.com/2018/04/pythondatasciencehandbook.pdf 5. Alex Holmes, Hadoop in practice, Dreamtech press.
6. Online References for data set http://archive.ics.uci.edu/ml/

https://www.kaggle.com/tanmoyie/us-graduate-schools-admission-parameters