## AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

## DEPARTMENT OF INFORMATION TECHNOLOGY ENGINEERING

## LABORATORY MANUAL

# 314457: DS & BDA Lab

| Sr. No. | Description |
|---|---|
| **I.** | **Institute and Department Vision, Mission, Quality Policy, Quality Objectives, PEOs, POs and PSOs** |
| **II.** | **List of Experiments** |
| | **Part A : Assignments based on the Hadoop** |
| 1 | Single node/Multiple node Hadoop Installation |
| 2 | 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. |
| 3 | Write an application using HiveQL for flight information system which will include<br>a. Creating, Dropping, and altering Database tables.<br>b. Creating an external Hive table.<br>c. Load table with data, insert new values and field in the table, Join tables with Hive<br>d. Create index on Flight Information Table<br>e. Find the average departure delay per day in 2008. |
| | **Part B : Assignments based on Data Analytics using Python** |
| 4 | Perform the following operations using Python on the Facebook metrics data sets a. Create data subsets<br>b. Merge Data<br>c. Sort Data<br>d. Transposing Data<br>e. Shape and reshape Data |

PREPARED BY APPROVED BY CONTROLLED COPY STAMP MASTER COPY STAMP

| | |
|---|---|
| 5 | Perform the following operations using Python on the Air quality and Heart Diseases data sets a. Data cleaning<br>b. Data integration<br>c. Data transformation<br>d. Error correcting |
| 6 | Integrate Python and Hadoop and perform the following operations on forest fire dataset a. Data analysis using the Map Reduce in PyHadoop<br>b. Data mining in Hive |
| 7 | Visualize the data using Python libraries matplotlib, seaborn by plotting the graphs for assignment no. 2 and 3 ( Group B) |
| 8 | Perform the following data visualization operations using Tableau on Adult and Iris datasets. a. 1D (Linear) Data visualization<br>b. 2D (Planar) Data Visualization<br>c. 3D (Volumetric) Data Visualization<br>d. Temporal Data Visualization<br>e. Multidimensional Data Visualization<br>f. Tree/ Hierarchical Data visualization<br>g. Network Data visualization |
| | **Part C : Model Implementation** |
| 9 | Create a review scrapper for any ecommerce website to fetch real time comments, reviews, ratings, comment tags, customer name using Python. |
| 10 | 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) |

# Vision and Mission of the Institute

## VISION:

To create opportunities for rural students to become able engineers and technocrats through continual excellence in engineering education.

## MISSION:

Our mission is to create self-disciplined, physically fit, mentally robust and morally strong engineers and technocrats with high degree of integrity and sense of purpose, who are capable to meet challenges of ever advancing technology for the benefit of mankind and nature.

We, the management and the faculty, therefore, promise to strive hard and commit ourselves to achieve this objective through a continuous process of learning and appreciation of needs of time.

# Vision and Mission of the Department

## Vision:

To transfer the rural learners into competent I.T. engineers and technocrats in emerging areas of I.T. Engineering education through continual excellence for the benefit of society.

## Mission:

**M-1:** To empower the youths in rural communities to be self-disciplined, physically fit, mentally robust and morally strong I.T. professionals.

**M-2:** To provides cutting-edge technical knowledge through continuous process in rapidly changing environment as per need of industry and surrounding world.

**M-3:** To provide opportunities for intellectual and personal growth of individuals in rural platform using high quality Information Technology education.

# PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

## To train the Information Technology students-

1. To develop competent I.T. graduate with knowledge of fundamental concepts In mathematics, science, engineering and ability to provide solution to complex engineering problem by analyzing , designing and designing using modern I.T. software and tools.
2. To prepare I.T. graduate with professional skills of better communication, teamwork to manage projects in I.T. field at global level and ability to conduct investigations of complex problems using research based knowledge and research methods.
3. To develop I.T graduates with ethical practices, societal contributions through communities understanding impact of professional engineering solutions in societal and environmental context and ability of lifelong learning.

# PROGRAMME SPECIFIC OUTCOMES (PSOs)

1. Apply principles of science, mathematics along with programming paradigms and problem solving skills using appropriate tools, techniques to expedite solution in I.T. domain.
2. Demonstrate core competencies related to I.T. in domain of Data structures & algorithms, Software Engineering & Modeling, Hardware, Distributed Computing, Networking & security, Databases, Discrete mathematics & algebra, Machine Learning, Operating System.
3. Demonstrate leadership qualities and professional skills in modern I.T. platform for creating innovative carrier paths in placements, entrepreneurship and higher studies

# PROGRAM OUTCOMES (POs)

**Engineering Graduates will be able to:**

- **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

- **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

# QUALITY POLICY

THE AMRUTVAHINI COLLEGE OF ENGINEERING IS COMMITTED TO DEVELOP IN YOUNG MINDS THE STATE – OF – THE – ART TECHNOLOGY AND HIGH ACADEMIC AMBIENCE BY SYNERGISING SPIRITUAL VALUES AND TECHNOLOGICAL COMPETENCE CONTINUALLY IN A LEARNING ENVIRONMENT.

# QUALITY OBJECTIVES

- To strive hard for academic excellence and synergizing spiritual & moral values.
- To improve overall development of student.
- To enhance industry-institute interaction.
- To provide assistance for placement & entrepreneurship development.
- To promote and encourage R&D activities.

**Aim: Single node/Multiple node Hadoop Installation**

**Objectives:** To learn Hadoop Installation on a) Single Node b) Multiple Node.

**Theory**:

Hadoop is written in Java, so you will need to have Java installed on your machine, version 8 or later. Sun's JDK is the one most widely used with Hadoop, although others have been reported to work. Hadoop runs on Unix and on Windows. Linux is the only supported production platform, but other flavors of Unix (including Mac OS X) can be used to run Hadoop for development. Windows is only supported as a development platform, and additionally requires Cygwin to run.During the Cygwin installation process, you should include the openssh package if you plan to run Hadoop in pseudo-distributed mode

**Procedure:**

**ALGORITHM**
**STEPS INVOLVED IN INSTALLING HADOOP IN STANDALONE MODE:-**
1. Command for installing ssh is **“sudo apt-get install ssh”**.
**2.** Command for key generation is **ssh-keygen –t rsa –P “ ”.**
**3.** Store the key into rsa.pub by using the command **cat $HOME/.ssh/id_rsa.pub** >> **$HOME/.ssh/authorized_keys**
**4.** Extract the java by using the command **tar xvfz jdk-8u60-linux-i586.tar.gz.**
5. Extract the eclipse by using the command **tar xvfz eclipse-jee-mars-R-linux-gtk.tar.gz**
6. Extract the hadoop by using the command **tar xvfz hadoop-2.9.0.tar.gz**

7. Move the java to **/usr/lib/jvm/** and eclipse to **/opt/** paths. Configure the java path in the eclipse.ini file
8. Export java path and hadoop path in ./bashrc
9. Check the installation successful or not by checking the java version and hadoop version
10. Check the hadoop instance in standalone mode working correctly or not by using an implicit hadoop jar file named as word count.
11. If the word count is displayed correctly in part-r-00000 file it means that standalone mode is installed successfully.
**ALGORITHM**

**STEPS INVOLVED IN INSTALLING HADOOP IN PSEUDO DISTRIBUTED MODE:-**
1. In order install pseudo distributed mode we need to configure the hadoop configuration files resides in the directory /home/lendi/hadoop-2.7.1/etc/hadoop.
2. First configure the hadoop-env.sh file by changing the java path.
3. Configure the core-site.xml which contains a property tag, it contains name and

value. Name as fs.defaultFS and value as hdfs://localhost:9000
4. Configure hdfs-site.xml.
5. Configure yarn-site.xml.
6. Configure mapred-site.xml before configure the copy mapred-site.xml.template to mapred-site.xml.
7. Now format the name node by using command hdfs namenode –format.
8. Type the command start-dfs.sh,start-yarn.sh means that starts the daemons like NameNode,DataNode,SecondaryNameNode ,ResourceManager,NodeManager.
9. Run JPS which views all daemons. Create a directory in the hadoop by using command hdfs dfs –mkdr /csedir and enter some data into lendi.txt using command nano lendi.txt and copy from local directory to hadoop using command hdfs dfs – copyFromLocal lendi.txt /csedir/and run sample jar file wordcount to check whether pseudo distributed mode is working or not.

10. Display the contents of file by using command hdfs dfs –cat /newdir/part-r-00000.

**MULTIPLE DISTRIBUTED MODE INSTALLATION:**

**ALGORITHM**

1. Stop all single node clusters
$stop-all.sh
2. Decide one as NameNode (Master) and remaining as DataNodes(Slaves).
3. Copy public key to all three hosts to get a password less SSH access
$ssh-copy-id –I $HOME/.ssh/id_rsa.pub lendi@l5sys24
4. Configure all Configuration files, to name Master and Slave Nodes.
$cd $HADOOP_HOME/etc/hadoop
$nano core-site.xml
$ nano hdfs-site.xml
5. Add hostnames to file slaves and save it.
$ nano slaves
6. Configure $ nano yarn-site.xml
7. Do in Master Node
$ hdfs namenode –format
$ start-dfs.sh
$start-yarn.sh
8. Format NameNode

9. Daemons Starting in Master and Slave Nodes
10. END

**it@IT-mmt-05:~$ start-all.sh**

```
it@IT-mmt-05:~$ jps
```

**Output:**

```
9026 NodeManager
7348 NameNode
9766 Jps
8887 ResourceManager
7507 DataNode
```

**Conclusion:**

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.

**Questions :**

## Q.1 Why is Hadoop important?

**Ans: Ability to store and process huge amounts of any kind of data, quickly.** With data volumes and varieties constantly increasing, especially from social media and the Internet of Things (IoT), that's a key consideration.

- ➢ **Computing power.** Hadoop's distributed computing model processes big data fast. The more computing nodes you use, the more processing power you have.
- ➢ **Fault tolerance.** Data and application processing are protected against hardware failure. If a node goes down, jobs are automatically redirected to other nodes to make sure the distributed computing does not fail. Multiple copies of all data are stored automatically.

- **Flexibility.** Unlike traditional relational databases, you don't have to preprocess data before storing it. You can store as much data as you want and decide how to use it later. That includes unstructured data like text, images and videos.
- **Low cost.** The open-source framework is free and uses commodity hardware to store large quantities of data.
- **Scalability.** You can easily grow your system to handle more data simply by adding nodes. Little administration is required.

**Q.2 What are the challenges of using Hadoop?**

**Ans: MapReduce programming is not a good match for all problems.** It's good for simple information requests and problems that can be divided into independent units, but it's not efficient for iterative and interactive analytic tasks. MapReduce is file-intensive. Because the nodes don't intercommunicate except through sorts and shuffles, iterative algorithms require multiple map-shuffle/sort-reduce phases to complete. This creates multiple files between MapReduce phases and is inefficient for advanced analytic computing.

2. **There's a widely acknowledged talent gap.** It can be difficult to find entry-level programmers who have sufficient Java skills to be productive with MapReduce. That's one reason distribution providers are racing to put relational (SQL) technology on top of Hadoop. It is much easier to find programmers with SQL skills than MapReduce skills. And, Hadoop administration seems part art and part science, requiring low-level knowledge of operating systems, hardware and Hadoop kernel settings.
3. **Data security.** Another challenge centers around the fragmented data security issues, though new tools and technologies are surfacing. The Kerberos authentication protocol is a great step toward making Hadoop environments secure.

| DS & BDA Lab | | | |
|---|---|---|---|
| **Experiment No: 2** | **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** | **Page** | **1/4** |

**Aim:** 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:** 1) To learn MapReduce concepts in Hadoop.

2) To learn a implementation of distributed in Hadoop .

**Theory:**

Log files contain list of actions that have been occurred whenever someone accesses to your website or web application. These log files resides in web servers. Each individual request is listed on a separate line in a log file, called a log entry. These log files fits very well with the MapReduce programming model making it a great example to understand the Hadoop Map/Reduce programming style. Our implementationconsists of three main parts:

1. Mapper
2. Reducer
3. Driver

**Procedure:**

**Step-1. Write a Mapper**

A Mapper overrides the —map function from the

Class "org.apache.hadoop.mapreduce.Mapper" which provides <key, value> pairs as the input. A Mapper implementation may output <key,value> pairs using the provided Context .

Input value of the Log file Map task will be a line of text from the input data file and the key would be the line number <line_number, line_of_text> . Map task outputs <word, one> for each word in the line of text.

**Pseudo-code**

```
void Map (key, value){
        for each log entry x in value:
        output.collect(x, 1);
}
```

### Step-2. Write a Reducer

A Reducer collects the intermediate <key,value> output from multiple map tasks and assemble a single result. Here, the Log file program will sum up the occurrence of each word to pairs as <logentry, occurrence>.

**Pseudo Code:**

```
void Reduce (key, <list of value>){
for each x in <list of value>:
sum+=x;
final_output.collect(key, sum);
}
```

### Step-3. Write Driver

The Driver program configures and run the MapReduce job. We use the main program to perform basic configurations such as:

□□Job Name : name of this Job<br>
□□Executable (Jar) Class: the main executable class. For here, Manifest.<br>
□□Mapper Class: class which overrides the "map" function. For here, Map.<br>
□□Reducer: class which override the "reduce" function. For here , Reduce.<br>
□□Output Key: type of output key. For here, Text.<br>
□□Output Value: type of output value. For here, IntWritable.<br>
□□File Input Path<br>
□□File Output Path

**INPUT:- Log file**

**Pseudo Code: NA**

**Output:**

```
hduser@cclab36-OptiPlex-3010:~/analyzelogs$ /usr/local/hadoop/bin/hdfs dfs -cat /output2000/part-00000
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
```

**Conclusion:** *MapReduce* is a programming framework that allows us to perform *distributed a*nd parallel processing on large data sets *in* a *distributed* environment. *MapReduce* consists of two distinct tasks – Map and Reduce. As the name *MapReduce* suggests, reducer phase takes place after mapper phase has been completed.

**Questions :**

### Q. 1 What Is MapReduce?

Ans: MapReduce is a program model for distributed computing that could be implemented in Java. The algorithm contains two key tasks, which are known as Map and Reduce.

The purpose of the Map task is to convert a dataset into another dataset where elements are broken down into key/value pairs known as tuples. TheReduce task combines those data tuples into a small set of tuples, using the output from a map as its input.

The distributed computing means splitting a task into several separate processes, which can then be carried out in parallel on large commodity hardware clusters. Once MapReduce has broken down the various elements of the large dataset into tuples and then further reduced them into a smaller set, the data that remains can be processed in parallel, which can significantly speed up the processing that needs to be carried out on the data.

**Q.2 Why is MapReduce important?**

Ans: MapReduce enables skilled programmers to write distributed applications without having to worry about the underlying distributed computing infrastructure. ... In short, this means MapReduce is now just one of many application frameworks you can use to develop and run applications on Hadoop.

| | | | |
|---|---|---|---|
| **DS & BDA Lab** | | | |
| **Experiment No: 3** | **Write an application using HiveQL for flight information system which will include**<br>**a. Creating, Dropping, and altering Database tables.**<br>**b. Creating an external Hive table.**<br>**c. Load table with data, insert new values and field in the table, Join tables with Hive**<br>**d. Create index on Flight Information Table**<br>**e. Find the average departure delay per day in 2008**. | **Page** | **1/10** |

**Aim:** 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**: To learn HiveQL database management system.

**Theory:**

**Hive : Hive** is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis.[2] Hive gives a SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop. Traditional SQL queries must be implemented in the MapReduce Java API to execute SQL applications and queries over distributed data. Hive provides the necessary SQL abstraction to integrate SQL-like queries (HiveQL) into the underlying Java without the need to implement queries in the low-level Java API. Since most data warehousing applications work with SQL-based querying languages, Hive aids portability of SQL-based applications to Hadoop.

**Pseudo Code:**

**Hive Installation Steps:-**

**Prerequisites** - Hadoop

```
sudo su hduser
cd
```

**--- download hive tar and copy to /home/hduser**

```
cp /home/student/Desktop/apache-hive-2.1.1.tar.gz  /home/hduser
```

| DS & BDA Lab | | | |
|---|---|---|---|
| **Experiment No: 3** | **Write an application using HiveQL for flight information system which will include**<br>**a. Creating, Dropping, and altering Database tables.**<br>**b. Creating an external Hive table.**<br>**c. Load table with data, insert new values and field in the table, Join tables with Hive**<br>**d. Create index on Flight Information Table**<br>**e. Find the average departure delay per day in 2008.** | **Page** | **2/10** |

```
tar -xvzf apache-hive-2.1.1.tar.gz
hduser1@student-HP-Pro-3330-MT:~$ ls
analogs                  examples.desktop     inp30      Music
apache-hive-2.1.1-bin        hbase-1.4.1-bin.tar.gz  inp40      Pictures
apache-hive-2.1.1-bin.tar.gz  inp                inp9       Public
Desktop                   inp1              input      Templates
Documents                  inp2              Manifest.txt  Videos
Downloads                  inp3              mapredu
```

1) **move hive to /usr/local/hive folder**

```
hduser1@student-HP-Pro-3330-MT:~$ sudo mv apache-hive-2.1.1-bin /usr/local/hive
[sudo] password for hduser1:

hduser1@student-HP-Pro-3330-MT:~$ pwd
/home/hduser1
hduser1@student-HP-Pro-3330-MT:~$ sudo gedit ~/.bashrc
export HIVE_HOME=/usr/local/hive
hduser1@student-HP-Pro-3330-MT:~$ source ~/.bashrc
```

2) **start hadoop**

```
hduser1@student-HP-Pro-3330-MT:~$ start-dfs.sh
18/03/20 13:22:24 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser1-namenode-student-HP-Pro-3330-MT.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser1-datanode-student-HP-Pro-3330-MT.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hduser1-secondarynamenode-student-HP-Pro-3330-MT.out
18/03/20 13:22:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

```
hduser1@student-HP-Pro-3330-MT:~$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser1-resourcemanager-student-HP-Pro-3330-MT.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser1-nodemanager-student-HP-Pro-3330-MT.out
```

```
hduser1@student-HP-Pro-3330-MT:~$ jps
7785 ResourceManager
7593 SecondaryNameNode
7930 NodeManager
8253 Jps
7389 DataNode
7245 NameNode
```

3) **create directories for hive on hadoop**

```
hduser1@student-HP-Pro-3330-MT:~$ cd /usr/local/hadoop

hduser1@student-HP-Pro-3330-MT:/usr/local/hadoop$ cd bin

hduser1@student-HP-Pro-3330-MT:/usr/local/hadoop/bin$ hdfs dfs -mkdir -p /user/hive/warehouse
18/03/20 13:24:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hduser1@student-HP-Pro-3330-MT:/usr/local/hadoop/bin$ hdfs dfs -mkdir -p /tmp/hive
18/03/20 13:24:36 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

4) **Give permissions for the directories for hive on hadoop**

```
hduser1@student-HP-Pro-3330-MT:/usr/local/hadoop/bin$ hdfs dfs -chmod 777 /tmp
```

```
18/03/20 13:33:09 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

hduser1@student-HP-Pro-3330-MT:/usr/local/hadoop/bin$ hdfs dfs -chmod 777 /use/hive/warehouse
18/03/20 13:33:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
chmod: `/use/hive/warehouse': No such file or directory

hduser1@student-HP-Pro-3330-MT:/usr/local/hadoop/bin$ hdfs dfs -chmod 777 /tmp/hive
18/03/20 13:34:17 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

5) **Initialize schematool**

```
hduser1@student-HP-Pro-3330-MT:/usr/local/hadoop/bin$ cd ..

hduser1@student-HP-Pro-3330-MT:/usr/local/hadoop$  cd ..

hduser1@student-HP-Pro-3330-MT:/usr/local$ cd hive

hduser1@student-HP-Pro-3330-MT:/usr/local/hive$ ls
bin   examples  jdbc  LICENSE  README.txt        scripts
conf  hcatalog  lib   NOTICE   RELEASE_NOTES.txt

hduser1@student-HP-Pro-3330-MT:/usr/local/hive$ cd bin
hduser1@student-HP-Pro-3330-MT:/usr/local/hive/bin$ ./schematool -initSchema -dbType derby
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
```

```
Metastore connection URL:    jdbc:derby:;databaseName=metastore_db;create=true
Metastore Connection Driver :          org.apache.derby.jdbc.EmbeddedDriver
Metastore connection User:    APP
Starting metastore schema initialization to 2.1.0
Initialization script hive-schema-2.1.0.derby.sql
Initialization script completed
schemaTool completed
```

```
hduser1@student-HP-Pro-3330-MT:/usr/local/hive/bin$ ls
beeline       ext       hive-config.cmd  hplsql       metatool
beeline.cmd  hive      hive-config.sh   hplsql.cmd    schematool
derby.log    hive.cmd  hiveserver2      metastore_db
```

6) **start Hive**

```
hduser1@student-HP-Pro-3330-MT:/usr/local/hive/bin$ ./hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-
2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-2.1.1.jar!/hive-
log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using
a different execution engine (i.e. spark, tez) or using Hive 1.X releases.

hive>
```

## Hbase (Nosql) single node installation:-

**Prerequisites** - Hadoop

1) **Login with hduser**

```
sudo su hduser1
```

```
cd
```

**2) start Hadoop**

```
start-dfs.sh
start-yarn.sh
jps
hadoop version
```

**3) download hbase tar 1.4.1, copy to desktop and copy to /home/hduser**

```
sudo cp hbase-1.4.1-bin.tar.gz /home/hduser/
```

**4)Extract the HBase**

```
tar -xvzf hbase-1.4.1.bin.tar.gz
ls
```

**5) move hbase to /usr/local/hbase folder**

```
sudo mv hbase-1.4.1 /usr/local/hbase
```

6) update bashrc

```
sudo gedit  ~/.bashrc
//.bashrc
export HBASE_HOME=/usr/local/hbase
export PATH=$PATH:$HBASE_HOME/bin
```

**7) check hosts folder**

```
sudo gedit  /etc/hosts
```

**8) update env in conf**

```
cd /usr/local/hbase/conf
sudo gedit hbase-env.sh
//hbase-env.sh
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
```

**9) upate hbase-site.xml**

```
sudo gedit hbase-site.xml
hbase-site.xml
-----------------
<property>
<name>hbase.rootdir</name>
<value>hdfs://localhost:9000/hbase</value>
</property>
```

```
<property>
<name>hbase.master.port</name>
<value>60001</value>
</property>

<property>
<name>hbase.cluster.distributed</name>
<value>true</value>
</property>

<property>
<name>hbase.zookeeper.property.dataDir</name>
<value>/usr/local/zookeeper</value>
</property>

<property>
<name>hbase.zookeeper.property.maxClientCnxns</name>
<value>35</value>
</property>
```

**10) remove zookeeper**

```
cd usr/local
 rm -fr zookeeper/
```

**11) edit regionservers**

```
cd /usr/local/hbase/conf
sudo gedit regionservers
```

**12) start hbase**

```
cd..
cd bin
./start-hbase.sh
```

Open your browser logon to
localhost:16010 (Hbase UI)

**13) start shell – table creation and data adding**

```
./hbase shell
hbase> create 'employee', 'saldet'
hbase>list
hbase>put 'employee' , '001', 'saldet:name', 'RMB'
hbase>scan 'employee'
hbase>put 'employee' , '001', 'saldet:loc', 'Nashik'
hbase>scan 'employee'
hbase>exit
```

**14) stop hbase**

```
./stop-hbase.sh
```

**15) stop all processes (hadoop)**

```
 stop-all.sh
```

**Output:**

**1) open hive**

```
hduser1@student-HP-Pro-3330-MT:/usr/local/hive/bin$ ./hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-2.1.1.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
```

**2) show databases**

```
hive> show databases;
OK
default
Time taken: 1.489 seconds, Fetched: 1 row(s)
```

**3) show tables**

```
hive> show tables;
OK
Time taken: 0.125 seconds
```

**4) create database db1**

```
hive> create database db1;
OK
Time taken: 0.745 seconds
```

**5) show cust table**

```
hive> create table cust(cname string, csal int) row format delimited fields terminated by ',' stored as textfile;
OK
Time taken: 0.862 seconds
hive> show tables;
OK
cust
Time taken: 0.055 seconds, Fetched: 1 row(s)
```

**6) load cust table from text file**

```
hive> load data local inpath '/home/student/Desktop/cust.txt' into table cust;
Loading data to table default.cust
OK
Time taken: 5.558 seconds
```

**7) select from cust table**

```
hive> select * from cust;
OK
pooja   20000
```

**Output :**

**1) Delete and drop table**

```
hbase(main):008:0> disable 'employee'
0 row(s) in 28.9140 seconds
hbase(main):009:0> drop 'employee'
```

```
0 row(s) in 4.5270 seconds
hbase(main):010:0> exists 'employee'
Table employee does not exist
0 row(s) in 0.0200 seconds
```

**Conclusion:** Apache Hive helps with querying and managing large datasets real fast. It is an ETL tool for Hadoop ecosystem. *HBase* is the Hadoop database, a distributed, scalable, big data store. Use Apache *HBase* when you need random, real time read/write access to your Big Data.

## *Questions:*

### *Q.1 List the advantages of Hive*

**Ans**: Hive was initially developed at Facebook to summarize, query, and analyze large amounts of data stored on a distributed file system. Hive makes it easy for non-programmers to read, write, and manage large datasets residing in distributed Hadoop storage using HiveQL SQL-like queries. Hive has gained a lot of popularity due to its ease of use and compatibility with existing business applications through ODBC.

| DS & BDA Lab | | | |
|---|---|---|---|
| **Experiment No: 4** | **Perform the following operations using Python on the Facebook metrics data sets<br>a. Create data subsets b. Merge Data c. Sort Data<br>d. Transposing Data e. Shape and reshape Data** | **Page** | **1/9** |

**Aim:.** 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 study data analysis operations using Python .

### Theory:

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

### Pseudo Code:

**titanic**

1. **Open jupyter notebook**

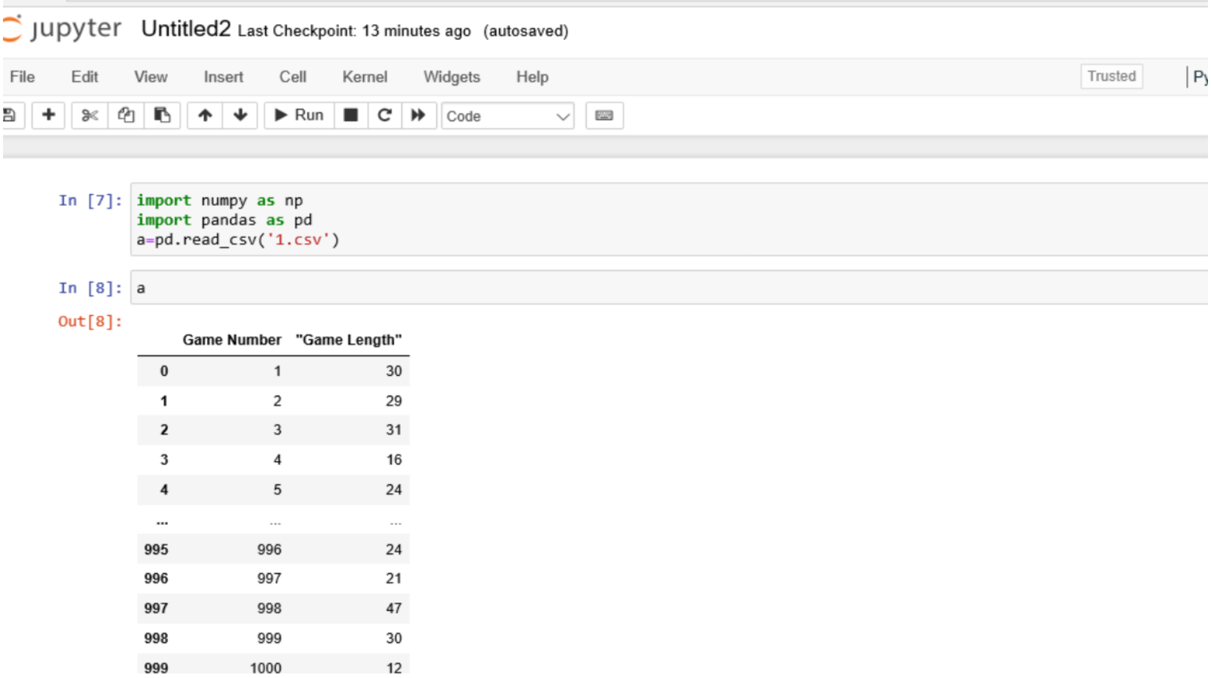
```
C:\Users\IT>jupyter notebook
```

2. **Upload CSV file**
3. **Create new file**

```
import numpy as np
import pandas as pd
a=pd.read_csv('1.csv')
```

4. **Display output**

```
a
```



```
In [7]: import numpy as np
        import pandas as pd
        a=pd.read_csv('1.csv')

In [8]: a

Out[8]:
        Game Number  "Game Length"
   0              1             30
   1              2             29
   2              3             31
   3              4             16
   4              5             24
 ...            ...            ...
 995            996             24
 996            997             21
 997            998             47
 998            999             30
 999           1000             12
```

SHIT + Enter --- > new line

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

## Creating a Dataframe

To create subsets of a dataframe, we need to create a dataframe

```
import pandas as pd
data = {"Roll-num": [10,20,30,40,50,60,70], "Age":[12,14,13,12,14,13,15],
"NAME":['John','Camili','Rheana','Joseph','Amanti','Alexa','Siri']}
block = pd.DataFrame(data)
print("Original Data frame:\n")
print(block)
```

**Output:**

```
Original Data frame:
   Roll-num  Age    NAME
0        10   12    John
1        20   14  Camili
2        30   13  Rheana
3        40   12  Joseph
4        50   14  Amanti
5        60   13   Alexa
6        70   15    Siri
```

## Create a subset of a Python dataframe using the loc() function

Python loc() function enables us to form a subset of a data frame according to a specific row or column or a combination of both.

**Syntax:**

```
pandas.dataframe.loc[]
```

**Example 1: Extract data of specific rows of a dataframe**

```
block.loc[[0,1,3]]
```

## Output:

```
Roll-num   Age NAME
0   10  12  John
1   20  14  Camili
3   40  12  Joseph
```

## Example 2: Create a subset of rows using slicing

```
block.loc[0:3]
```

## Output:

```
Roll-num   Age NAME
0   10  12  John
1   20  14  Camili
2   30  13  Rheana
3   40  12  Joseph
```

## Example 3: Create a subset of particular columns using labels

```
block.loc[0:2,['Age','NAME']]
```

## Output:

```
Age NAME
0   12  John
1   14  Camili
2   13  Rheana
```

### 2. Using Python iloc() function to create a subset of a dataframe

Python iloc() function enables us to create subset choosing specific values from rows and columns based on indexes.

That is, unlike loc() function which works on labels, iloc() function works on index values. We can choose and create a subset of a Python dataframe from the data providing the index numbers of the rows and columns.

**Syntax:**

```
pandas.dataframe.iloc[]
```

**Example:**

```
block.iloc[[0,1,3,6],[0,2]]
```

Here, we have created a subset which includes the data of the rows 0,1,3 and 6 as well as column number 0 and 2 i.e. ‘Roll-num’ and ‘NAME’.

**Output:**

```
Roll-num    NAME
0   10  John
1   20  Camili
```

```
3   40  Joseph
6   70  Siri
```

### 3. Indexing operator to create a subset of a dataframe

In a simple manner, we can make use of an indexing operator i.e. square brackets to create a subset of the data.

**Syntax:**

```
dataframe[['col1','col2','colN']]
```

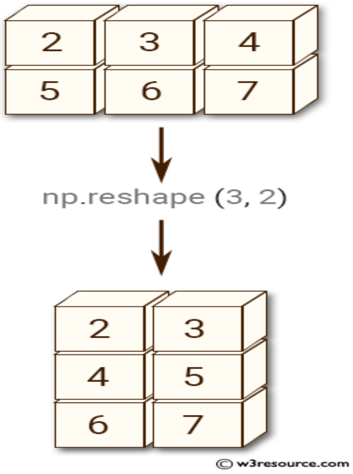
### Shape

The *shape* attribute shows the number of items in each dimension. Checking a DataFrame's shape returns a tuple with two integers. The first is the number of rows and the second is the number of columns.

df_hurricanes.shape(3, 2)

numpy.reshape

Gives a new shape to an array without changing its data.



```
1 ) import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(4, 3)

print(newarr)
```

```
2) import numpy as np

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

newarr = arr.reshape(2, 3, 2)

print(newarr)
```

```
 import numpy as np
x=np.arange(12)
y=np.reshape(x, (4,3))
x
y
```

```
#read the ".csv" file into dataframe "df"
 df = pd.read_csv('50_Startups.csv')

 #creating subsets and concat
 df2 = df[:25]
 df3 = df[25:]

 #joining subsets horizontally
 df6 = pd.concat([df2,df3],axis=0)

 df4 = df.iloc[:,:2]
 df5 = df.iloc[:,2:]

 #joining subsets vertically
 df6 = pd.concat([df4,df5],axis=1)

 #merge
 df2 = df.iloc[:,:3]
 df3 = df.iloc[:,2:]

 df7 = df2.merge(df3, on=['Marketing Spend'],how='inner')

 #sort
 x = df.sort_values(by = ['Marketing Spend'],ascending=True)

 #transpose
 x = df.T

 #Casting data to wide format
 df = pd.read_csv('abc.txt', sep=" ", names=['date', 'name', 'dollars'])

 df2 = df.pivot(index='date', columns='name', values='dollars').reset_index()

 #Melting Data to long format
 df3 = pd.melt(df2,id_vars='date',value_vars=['George','Lisa','Michael'])
```

**Conclusion :** Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional.

**Questions**

## Q.1 Why learn Python for data analysis?

Ans: Python has gathered a lot of interest recently as a choice of language for data analysis.

Following are the t some reasons which go in favor of learning Python:

- Open Source – free to install
- Awesome online community
- Very easy to learn
- Can become a common language for data science and production of web based analytics products.

## Q. 2 How to install Python?

Ans : There are 2 approaches to install Python:

- Download Python directly from its project site and install individual components and libraries.
- Alternately, we can download and install a package, which comes with pre-installed libraries or downloading Anaconda.

| DS & BDA Lab | | | |
|---|---|---|---|
| **Experiment No: 5** | **Perform the following operations using Python on the Air quality and Heart Diseases data sets a. Data cleaning b. Data integration c. Data transformation d. Error correcting e. Data model building** | **Page** | **1/8** |

**Aim:** Perform the following operations using Python on the Air quality and Heart Diseases data sets

a. Data cleaning

**b.** Data integration

**c.** Data transformation

**d.** Error correcting

**e.** Data model building

### Objectives:

1. To study data analysis operations using Python .

### Theory:

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

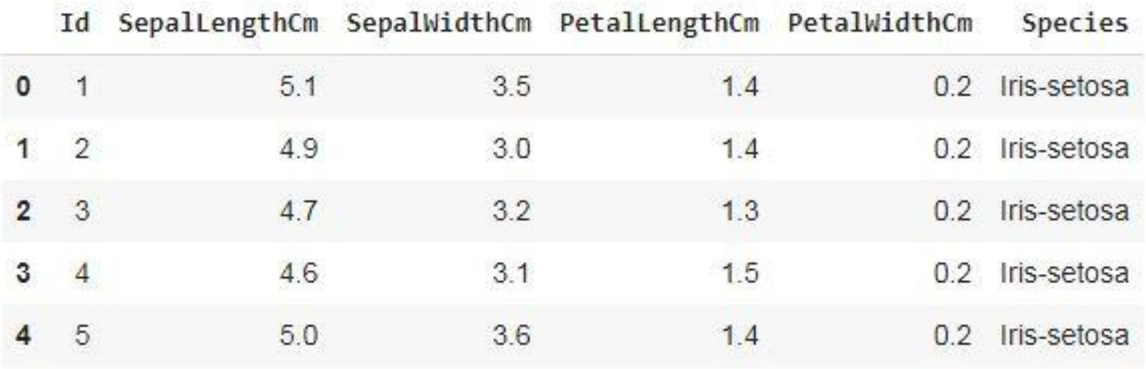
### Pseudo Code:

```
#importing module
import pandas as pd
```

### Import Dataset

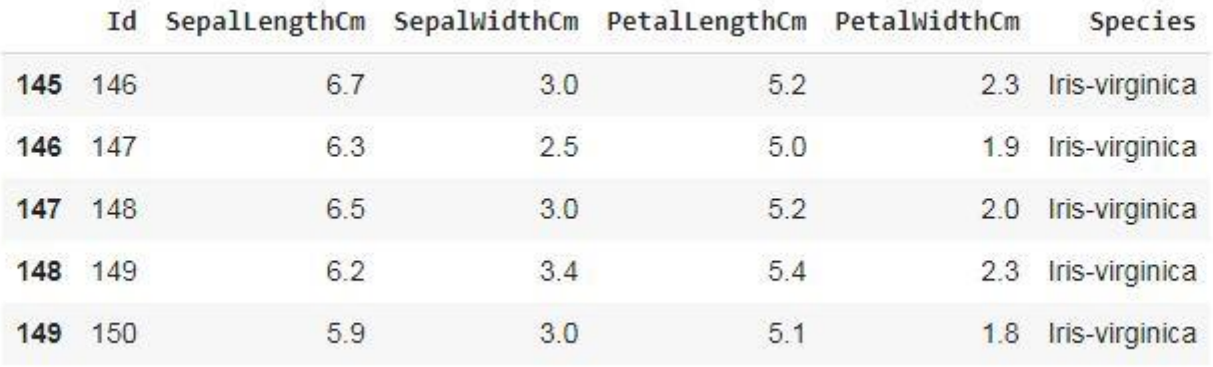
To import the dataset we use the read_csv() function of pandas and store it in the DataFrame named as data. As the dataset is in tabular format, when working with tabular data in Pandas it will be automatically converted in a DataFrame. DataFrame is a two-dimensional, mutable data structure in

Python. It is a combination of rows and columns like an excel sheet.
#importing the dataset by reading the csv file
data = pd.read_csv(Iris.csv)
#displaying the first five rows of dataset
data.head()



| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|---|---|---|---|---|---|---|
| 0 | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 1 | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| 2 | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| 3 | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| 4 | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |

The head() function is a built-in function in pandas for the dataframe used to display the rows of the dataset. We can specify the number of rows by giving the number within the parenthesis. By default, it displays the first five rows of the dataset. If we want to see the last five rows of the dataset we use the tail()function of the dataframe like this:
#displayinf last five rows of dataset
data.tail()



| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|---|---|---|---|---|---|---|
| 145 | 146 | 6.7 | 3.0 | 5.2 | 2.3 | Iris-virginica |
| 146 | 147 | 6.3 | 2.5 | 5.0 | 1.9 | Iris-virginica |
| 147 | 148 | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica |
| 148 | 149 | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |
| 149 | 150 | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica |

**Rebuild Missing Data**

To find and fill the missing data in the dataset we will use another function. There are 4 ways to find the null values if present in the dataset. Let’s see them one by one:

**Using isnull() function:**
data.isnull()

This function provides the boolean value for the complete dataset to know if any null value is present or not.

**Using isna() function:**
**data.isna()**

This is the same as the isnull() function. Ans provides the same output.

**Using isna().any()**
**data.isna().any()**

This function also gives a boolean value if any null value is present or not, but it gives results column-wise, not in tabular format.

**Using isna(). sum()**
**data.isna().sum()**

**Data integration**
**Install scipy**
**C:\Users\IT>pip install scipy**

**Check its version on IDE**
**>>print(scipy.__version__)**

```
import scipy.integrate as integrate
>>> import scipy.special as special
>>> result = integrate.quad(lambda x: special.jv(2.5,x), 0, 4.5)
>>> result
```

```
from numpy import sqrt, sin, cos, pi
>>> I = sqrt(2/pi)*(18.0/27*sqrt(2)*cos(4.5) - 4.0/27*sqrt(2)*sin(4.5) +
...              sqrt(2*pi) * special.fresnel(3/sqrt(pi))[0])
from scipy.integrate import quad
>>> def integrand(x, a, b):
...     return a*x**2 + b
...
>>> a = 2
>>> b = 1
>>> I = quad(integrand, 0, 1, args=(a,b))
>>> I
```

**C:\Users\IT>pip install plotly**

## Data Integration

Data Integration is a data preprocessing technique that involves combining data from multiple heterogeneous data sources into a coherent data store and provide a unified view of the data. These sources may include multiple data cubes, databases, or flat files.

○

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

plt.style.use('ggplot')

df = pd.read_csv('train.csv')

df.head()


df.drop(['PassengerId','Ticket','Name'],inplace=True,axis=1)
```

**# Remove duplicate rows of the dataframe using carb variable**

```
distinct(mydata,carb, .keep_all= TRUE)
```

**Data integration**

Union and union_all Function in R using Dplyr (union of data frames):

UNION function in R combines all rows from both the tables and removes duplicate records from the combined dataset

**Create two data frames**

```
library(dplyr)
 #  union two dataframes  without duplicates
union(df1,df2)
df1 = data.frame(CustomerId = c(1:6), Product = c(rep("Oven", 3), rep("Television", 3)))
df2 = data.frame(CustomerId = c(4:7), Product = c(rep("Television", 2), rep("Air conditioner", 2)))
union_all(tab1,tab2)
union(tab1,tab2)
```

**Data transformation**

Transformationsin R"Data transformation" is a fancy term for changing the values of observations through some mathematical operation.

```
data12 <-sqrt(tab1$Prepaid_New)
> data13 <-sqrt(tab1$Prepaid_New)^2
> xyz <- grep("Television")
Error in grep("Television") : argument "x" is missing, with no default
> xyz <- grep("Television",dff)
> xyz
[1] 2
> plot(example$Solar.R, type="o", col="blue")
> plot(example, type="o", col="blue")
Error in plot.default(...) :
  formal argument "type" matched by multiple actual arguments
> boxplot(example)
> boxplot(example$Temp)
> barplot(airquality$Wind)
> barplot(airquality$Wind,width = 3)
> barplot(airquality$Wind,width = 5)
> barplot(airquality$Wind,width = 15)
setwd("D:/DSBDA/Big data workshop/BDS/R_LAB");
tab1=read.csv("data_h.csv")
```

**Error correcting**

```
#find mean of prepaid column
Meanprepaidconnection=mean(tab1$Prepaid_New)
Meanprepaidconnection
#Put one value NA in Prepaid new column
xx=edit(tab1)
Meanprepaidconnection=mean(xx$ Prepaid_New)
Meanprepaidconnection
# now result is [1] NA since one value is missing
#to remove this missing value for taking mean of prepaid column
Meanprepaidconnection=mean(xx$Prepaid_New,na.rm = TRUE)
Meanprepaidconnection
# now result is 45.39
person=c(1, 2, NA)
person
xy=na.omit(person)
xy
#The result of the na.omit function is adata.frame where incomplete rows have been deleted.
#The row.names of the removed records are stored in an attribute called na.action
#violatedEdits returns a logical array indicating for each row of the data, which rules are violated.
```

**Data model building**

Pattern Matching and Replacement

grep, grepl, regexpr, gregexpr and regexec search for matches to argument pattern within each element of a character vector: they differ in the format of and amount of detail in the results.

Approximate String Matching (Fuzzy Matching)

Searches for approximate matches to pattern (the first argument) within each element of the string x (the second argument) using the generalized Levenshtein edit distance (the minimal possibly weighted number of insertions, deletions and substitutions needed to transform one string into another).

```
aa=readLines("data_p.csv")
aa
xyz1=grepl("^%", aa)
(dat <- aa[!xyz1])
```

**Example of Data model building**

```
> data_h <- read.csv("~/jugal/data_h.csv")
>   View(data_h)
> nomonth=grepl('Feb',data_h$Month)
> nomonth
```

```
 [1] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE
[14]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

(False value for 'Feb' month)

```
> newdata=data_h$Month[!nomonth]
> newdata
 [1] Jan Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Mar Apr May Jun Jul Aug Sep Oct
[21] Nov Dec
Levels: Apr Aug Dec Feb Jan Jul Jun Mar May Nov Oct Sep
>
```

(Grepl command remove the 'Feb' month data and give rest of the data like Pattern Matching and Replacement)

**Output: --**

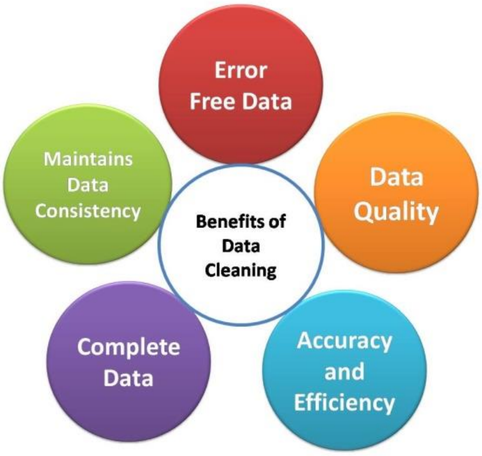
**Conclusion:** Python  is a great tool to explore and investigate the data. Elaborate analysis like clustering, correlation, and data reduction are done with Python . This is the most crucial part, without a good feature engineering and model, the deployment of the machine learning will not give meaningful results.

**Q. 1 What is data cleaning?**

Ans : When working with multiple data sources, there are many chances for data to be incorrect, duplicated, or mislabeled. If data is wrong, outcomes and algorithms are unreliable, even though they may look correct. *Data cleaning* is the process of changing or eliminating garbage, incorrect, duplicate, corrupted, or incomplete data in a dataset. There's no such absolute way to describe the precise steps in the data cleaning process because the processes may vary from dataset to dataset. Data cleansing, data cleansing, or data scrub is that the initiative among the general data preparation process. Data cleaning plays an important part in developing reliable answers and within the analytical process and is observed to be a basic feature of the info science basics. The motive of data cleaning services is to construct uniform and standardized data sets that enable data analytical tools and business intelligence easy access and perceive accurate data for each problem.

**Q.2 Why data cleaning is essential?**

Ans: Data cleaning is the most important task that should be done as a data science professional. Having wrong or bad quality data can be detrimental to processes and analysis. Having clean data will ultimately increase overall productivity and permit the very best quality information in your decision-making. Following are some reasons why data cleaning is essential:



| DS & BDA Lab | | | |
|---|---|---|---|
| **Experiment No: 6** | **Integrate Python and Hadoop and perform the following operations on forest fire dataset**<br>**a. Data analysis using the Map Reduce in PyHadoop**<br>**b. Data mining in Hive** | **Page** | **1/3** |

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

**a.** Data analysis using the Map Reduce in PyHadoop

**b.** Data mining in Hive

**Objectives**: To study the Integrate Python and Hadoop

**Theory:**

**Hadoop Streaming :**

• Hadoop streaming is a Hadoop utility for running the Hadoop MapReduce job with executable scripts such as Mapper and Reducer.

• This is similar to the pipe operation in Linux.

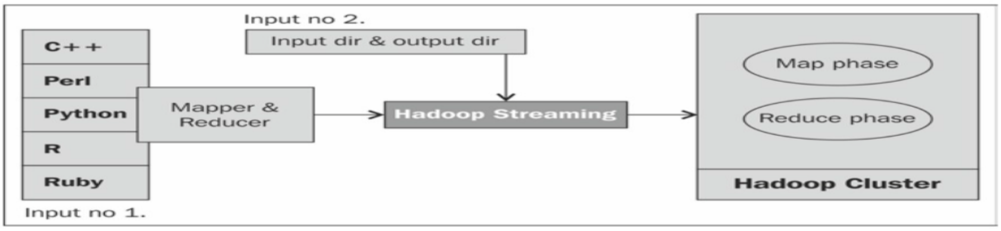
• With this, the text input file is printed on stream ( stdin ), which is provided as an input to Mapper and the output ( stdout ) of Mapper is provided as an input to Reducer; finally, Reducer writes the output to the HDFS directory.

• The main advantage of the Hadoop streaming utility is that it allows Java as well as non-Java programmed MapReduce jobs to be executed over Hadoop clusters.

• Also, it takes care of the progress of running MapReduce jobs.

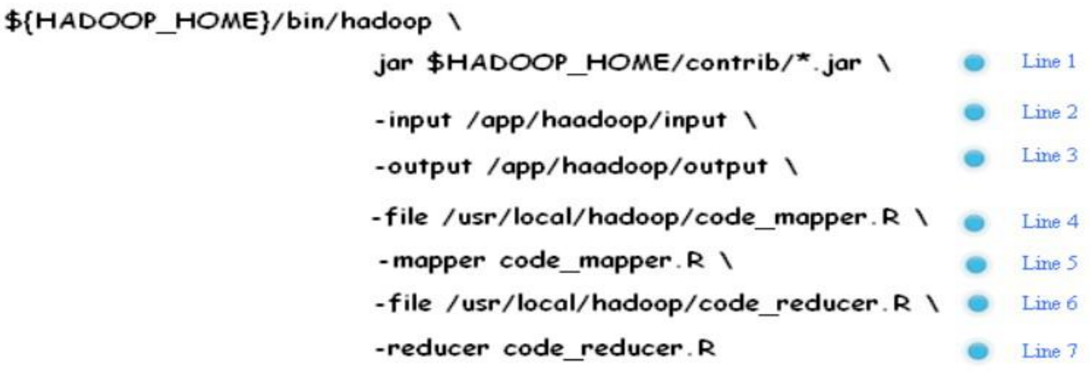
• The Hadoop streaming supports the Perl, Python, PHP, R, and C++ programming languages.

• To run an application written in other programming languages, the developer just needs to translate the application logic into the Mapper and Reducer sections with the key and value output elements.



**Pseudo Code:**

**Hadoop Streaming Command :**



```
${HADOOP_HOME}/bin/hadoop \
                    jar $HADOOP_HOME/contrib/*.jar \
                    -input /app/haadoop/input \
                    -output /app/haadoop/output \
                    -file /usr/local/hadoop/code_mapper.R \
                    -mapper code_mapper.R \
                    -file /usr/local/hadoop/code_reducer.R \
                    -reducer code_reducer.R
```

**Output: --**

**Conclusion:**

Python  can communicate with the other language. It is possible to call Python, Java, C++ in R. The world of big data is also accessible to Python . You can connect Python  with different databases like Spark or Hadoop.

**Questions**

**Q.1 Suppose that I want to know the values in c(1, 2, 6, 3, 19) that are not present in c(2, 6, 14, 3, 15). How can you carry this out using built-in function as well as without it?**

Ans: There are two methods to execute this problem –

- By using setdiff() function – setdiff(c(1, 2, 6, 3, 19), c(2, 6, 14, 3, 15)) and,
- Through the %in% as – c(1, 4, 5, 9, 10)[!c(1, 4, 5, 9, 10) %in% c(1, 5, 10, 11, 13)

**Q.2 Is it possible to plot all the features of a dataset at once? If so, how?**

**Ans.**

1. > library(tabplot) #DataFlair
2. > tableplot(iris)

| **DS & BDA Lab** | | | |
|---|---|---|---|
| **Experiment No: 7** | **Visualize the data using Python libraries matplotlib, seaborn by plotting the graphs for assignment no. 2 and 3 ( Group B)** | **Page** | **1/9** |

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

**Objectives:** To Study Data Visualization with Python

**Theory:**

With ever increasing volume of data in today’s world, it is impossible to tell stories without these visualizations. While there are dedicated tools like Tableau, QlikView and d3.js, nothing can replace a modeling / statistics tools with good visualization capability. It helps tremendously in doing any exploratory data analysis as well as feature engineering. This is where Python offers incredible help.

Python Programming offers a satisfactory set of inbuilt function and libraries (such as ggplot2, leaflet, lattice) to build visualizations and present data.

Following are different Charts and Graphs supported in Python language for Visualize the data :

- Pie chart
- Bar chart
- Box plots
- Historams
- Line graphs
- Scatter plots

**Using Matplotlib**

1) import matplotlib.pyplot as plt

a = [1,2,3,4,5,6,5,4,3,2,1]

plt.plot(a)

2) import matplotlib.pyplot as plt

a = [1,2,3,4,5,6,5,4,3,2,1]

```
b = [10,20,30,40,50,60,50,40,30,20,10]

plt.plot(a,b)

3) import matplotlib.pyplot as plt


a = [1,2,3,4,5,6,5,4,3,2,1]


b = [10,20,30,40,50,60,50,40,30,20,10]


plt.plot(a,b)


plt.xlabel('year')

plt.ylabel('Yield(tones per hector)')


a= [1,2,3,4,5,6,5,4,3,2,1]

Plt.plot(a)
```

Basic Example of plotting Graph

```
1) from matplotlib import pyplot as plt
#ploting our canvas
plt.plot([1,2,3],[4,5,1])
#display the graph
plt.show()

2) from matplotlib import pyplot as plt
```

```
x = [1,2,3]
y = [10,11,12]

plt.plot(x,y)
plt.title("Line graph")
plt.ylabel('Y axis')
plt.xlabel('X axis')
plt.show()
```

### Bar chart using Matplotlib

```
import seaborn as sns
#Creating the dataset
df = sns.load_dataset('titanic')
df=df.groupby('who')['fare'].sum().to_frame().reset_index()
#Creating the bar chart
plt.barh(df['who'],df['fare'],color = ['#F0F8FF','#E6E6FA','#B0E0E6'])

#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
#Show the plot
plt.show()
```

### Bar chart using Seaborn

```
import seaborn as sns
#Creating bar plot
sns.barplot(x = 'fare',y = 'who',data = df ,palette = "Blues")
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```

*Column chart using Matplotlib*

```
import seaborn as sns
#Creating the dataset
df = sns.load_dataset('titanic')
df=df.groupby('who')['fare'].sum().to_frame().reset_index()
#Creating the column plot
plt.bar(df['who'],df['fare'],color = ['#F0F8FF','#E6E6FA','#B0E0E6'])
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
#Show the plot
plt.show()
```

Column chart using Seaborn

```
#Reading the dataset
titanic_dataset = sns.load_dataset('titanic')
#Creating column chart
sns.barplot(x = 'who',y = 'fare',data = titanic_dataset,palette = "Blues")
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```

Grouped bar chart

A grouped bar chart is used when we want to compare the values in certain groups and sub-groups

*Grouped bar chart using Matplotlib*

```
#Creating the dataset
df = sns.load_dataset('titanic')
df_pivot = pd.pivot_table(df, values="fare",index="who",columns="class", aggfunc=np.mean)
#Creating a grouped bar chart
ax = df_pivot.plot(kind="bar",alpha=0.5)
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
```

```
# Show the plot
plt.show()
```

*Grouped bar chart using Seaborn*

```
#Reading the dataset
titanic_dataset = sns.load_dataset('titanic')
#Creating the bar plot grouped across classes
sns.barplot(x = 'who',y = 'fare',hue = 'class',data = titanic_dataset, palette = "Blues")
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```

**Stacked bar chart**

A stacked bar chart is used when we want to compare the total sizes across the available groups and the composition of the different sub-groups

*Stacked bar chart using Matplotlib*

```
# Stacked bar chart
#Creating the dataset
df = pd.DataFrame(columns=["A","B", "C","D"],
                  data=[["E",0,1,1],
                        ["F",1,1,0],
                        ["G",0,1,0]])

df.plot.bar(x='A', y=["B", "C","D"],  stacked=True,  width = 0.4,alpha=0.5)
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
#Show the plot
plt.show()
```

*Stacked bar chart using Seaborn*

```
dataframe = pd.DataFrame(columns=["A","B", "C","D"],
                  data=[["E",0,1,1],
                        ["F",1,1,0],
   ["G",0,1,0]])
dataframe.set_index('A').T.plot(kind='bar', stacked=True)
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```

**Line chart**

A line chart is used for the representation of continuous data points. This visual can be effectively utilized when we want to understand the trend across time.

*Line chart using Matplotlib*

```
#Creating the dataset
df = sns.load_dataset("iris")
df=df.groupby('sepal_length')['sepal_width'].sum().to_frame().reset_index()
#Creating the line chart
plt.plot(df['sepal_length'], df['sepal_width'])
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
#Show the plot
plt.show()
```

*Line chart using Seaborn*

```
#Creating the dataset
cars = ['AUDI', 'BMW', 'NISSAN',
        'TESLA', 'HYUNDAI', 'HONDA']
data = [20, 15, 15, 14, 16, 20]
#Creating the pie chart
plt.pie(data, labels = cars,colors = ['#F0F8FF','#E6E6FA','#B0E0E6','#7B68EE','#483D8B'])
#Adding the aesthetics
plt.title('Chart title')
#Show the plot
plt.show()
```

### Pie chart

Pie charts can be used to identify proportions of the different components in a given whole.

*Pie chart using Matplotlib*

```
#Creating the dataset
cars = ['AUDI', 'BMW', 'NISSAN',
        'TESLA', 'HYUNDAI', 'HONDA']
data = [20, 15, 15, 14, 16, 20]
#Creating the pie chart
plt.pie(data, labels = cars,colors = ['#F0F8FF','#E6E6FA','#B0E0E6','#7B68EE','#483D8B'])
#Adding the aesthetics
plt.title('Chart title')
#Show the plot
plt.show()
```

### Area chart

Area charts are used to track changes over time for one or more groups. Area graphs are preferred over line charts when we want to capture the changes over time for more than 1 group.

*Area chart using Matplotlib*

```
#Reading the dataset
x=range(1,6)
y=[ [1,4,6,8,9], [2,2,7,10,12], [2,8,5,10,6] ]
#Creating the area chart
ax = plt.gca()
ax.stackplot(x, y, labels=['A','B','C'],alpha=0.5)
#Adding the aesthetics
plt.legend(loc='upper left')
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
#Show the plot
```

```
plt.show()
```

*Area chart using Seaborn*

```
# Data
years_of_experience =[1,2,3]
salary=[ [6,8,10], [4,5,9], [3,5,7] ]
# Plot
plt.stackplot(years_of_experience,salary, labels=['Company A','Company B','Company C'])
plt.legend(loc='upper left')
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```

**Questions**

Q.1 Why learn Python for data analysis?

Ans: Python has gathered a lot of interest recently as a choice of language for data analysis. Following are the t some reasons which go in favor of learning Python:

- Open Source – free to install
- Awesome online community
- Very easy to learn
- Can become a common language for data science and production of web based analytics products.

## Q. 2 How to install Python?

Ans : There are 2 approaches to install Python:

- Download Python directly from its project site and install individual components and libraries.
- Alternately, we can download and install a package, which comes with pre-installed libraries or downloading Anaconda.

| **DS & BDA Lab** | | | |
|---|---|---|---|
| **Experiment No: 8** | Perform the following data visualization operations using Tableau on Adult and Iris datasets. **a.** 1D (Linear) Data visualization **b.** 2D (Planar) Data Visualization **c.** 3D (Volumetric) Data Visualization **d.** Temporal Data Visualization **e.** Multidimensional Data Visualization **f.** Tree/ Hierarchical Data visualization **g.** Network Data visualization | **Page** | **1/3** |

**Aim:** Perform the following data visualization operations using Tableau on Adult and Iris datasets.

**a.** 1D (Linear) Data visualization

**b.** 2D (Planar) Data Visualization

**c.** 3D (Volumetric) Data Visualization

**d.** Temporal Data Visualization

**e.** Multidimensional Data Visualization

**f.** Tree/ Hierarchical Data visualization

**g.** Network Data visualization

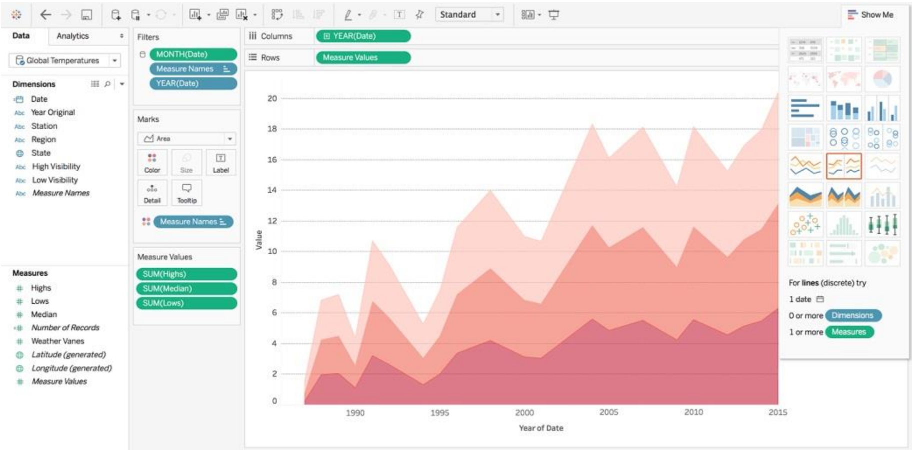
**Objectives**: To study the data visualization using Tableau tool

**Theory:**

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.

**Pseudo Code: NA**

**Output:**



**Conclusion:** Tableau software is used to translate queries into visualization.It is also used for managing metadata. Tableau software imports data of all sizes and ranges.

**Questions**

### Q. 1 What is Tableau?

Ans: Tableau is a powerful and fastest growing data visualization tool used in the Business Intelligence Industry. It helps in simplifying raw data into the very easily understandable format.

Data analysis is very fast with Tableau and the visualizations created are in the form of dashboards and worksheets. The data that is created using Tableau can be understood by professional at any level in an organization. It even allows a non-technical user to create a customized dashboard.

The best feature Tableau are

- Data Blending

- Real time analysis
- Collaboration of data

The great thing about Tableau software is that it doesn't require any technical or any kind of programming skills to operate. The tool has garnered interest among the people from all sectors such as business, researchers, different industries, etc.

### Q.2 Tableau Product Suite

Ans: The Tableau Product Suite consists of

- Tableau Desktop
- Tableau Public
- Tableau Online
- Tableau Server
- Tableau Reader

<table>
<tr><td colspan="4"><b>DS &amp; BDA Lab</b></td></tr>
<tr><td><b>Experiment No: 9</b></td><td><b>Create a review scrapper for any ecommerce website to fetch real time comments, reviews, ratings, comment tags, customer name using Python</b>.</td><td><b>Page</b></td><td><b>1/4</b></td></tr>
</table>

| DS & BDA Lab | | | |
|---|---|---|---|
| **Experiment No: 10** | **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)** . | **Page** | **1/1** |