

**INDEX**

<b>Sr. no</b>	<b>Aim</b>	<b>Date</b>	<b>Page No</b>	<b>Sign</b>
<b>1</b>	<b>Install, configure and run Hadoop and HDFS and explore HDFS.</b>		<b>2</b>	
<b>2</b>	<b>Implement word count / frequency programs using MapReduce.</b>		<b>9</b>	
<b>3</b>	<b>Implement an application that stores big data in Hbase / MongoDB and manipulate it using R / Python.</b>		<b>13</b>	
<b>4</b>	<b>Implement Decision tree classification techniques.</b>		<b>20</b>	
<b>5</b>	<b>Implement SVM classification techniques.</b>		<b>21</b>	
<b>6</b>	<b>REGRESSION MODEL: Import data from web storage. Perform Logistic Regression to find the relationship between variables affecting student admission based on GRE score, GPA, and student rank. Check model fit using require(foreign) and require(MASS).</b>		<b>24</b>	
<b>7</b>	<b>MULTIPLE REGRESSION MODEL: Apply multiple regressions on the above dataset, if data has a continuous independent variable.</b>		<b>26</b>	
<b>8</b>	<b>CLASSIFICATION MODEL: a. Install relevant package for classification. b. Choose classifier for classification problem. c. Evaluate the performance of classifier.</b>		<b>28</b>	
<b>9</b>	<b>CLUSTERING MODEL: a. Clustering algorithms for unsupervised classification. b. Plot the cluster data using R visualizations.</b>		<b>30</b>	

# Practical 1

## Install, configure and run Hadoop and HDFS and explore HDFS.

### Steps to Install Hadoop

1. Install Java JDK 1.8
2. Download Hadoop and extract and place under C drive
3. Set Path in Environment Variables
4. Config files under Hadoop directory
5. Create folder datanode and namenode under data directory
6. Edit HDFS and YARN files
7. Set Java Home environment in Hadoop environment
8. Setup Complete. Test by executing start-all.cmd

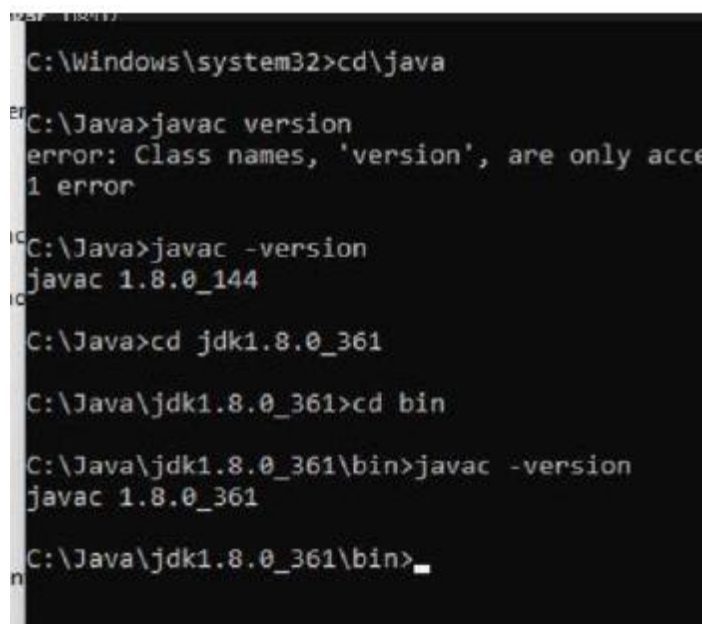
There are two ways to install Hadoop, i.e.

9. Single node
10. Multi node

Here, we use multi node cluster.

### 1. Install Java

- – Java JDK Link to download  
o <https://www.oracle.com/java/technologies/javase-jdk8-downloads.html>
- – extract and install Java in C:\Java
- – open cmd and type -> javac -version

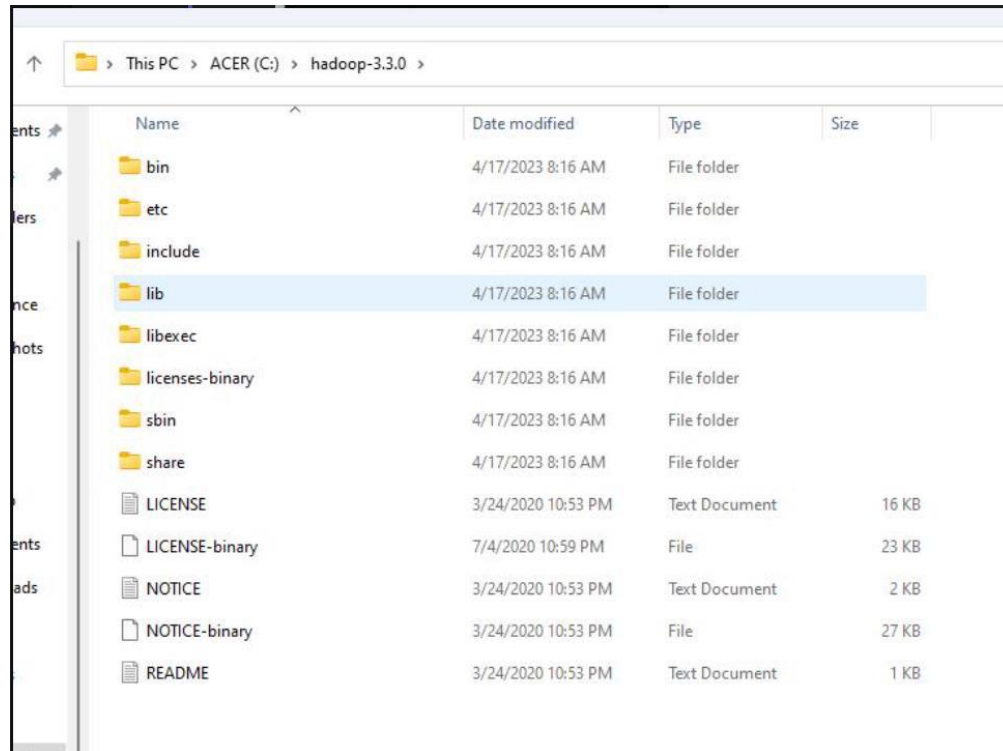


```
C:\Windows\system32>cd\java
C:\Java>javac version
error: Class names, 'version', are only accepted if used by javac
1 error
C:\Java>javac -version
javac 1.8.0_144
C:\Java>cd jdk1.8.0_361
C:\Java\jdk1.8.0_361>cd bin
C:\Java\jdk1.8.0_361\bin>javac -version
javac 1.8.0_361
C:\Java\jdk1.8.0_361\bin>
```

## 2. Download Hadoop

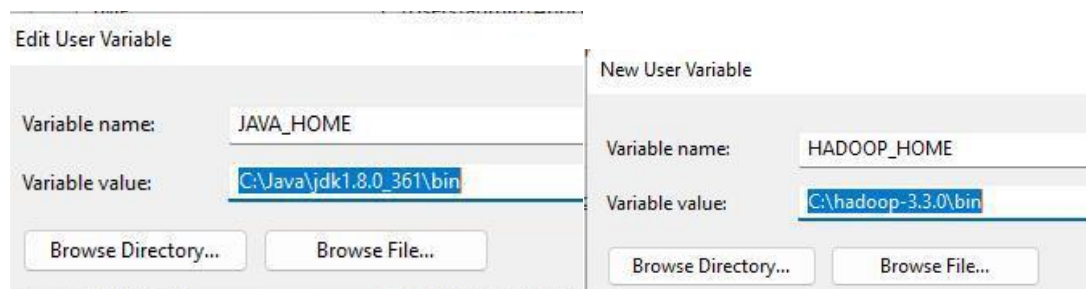
<https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz>

- right click .rar.gz file -> show more options -> 7-zip->and extract to C:\Hadoop-3.3.0\



## 3. Set the path JAVA\_HOME Environment variable

4. Set the path HADOOP\_HOME Environment variable Click on New to both user variables and system variables. Click on user variable -> path -> edit-> add path for Hadoop and java upto 'bin'



Click Ok, Ok, Ok.

## 5. Configurations

**Edit file C:/Hadoop-3.3.0/etc/hadoop/core-site.xml,**

paste the xml code in folder and save

```
<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>
```

=====

**Rename “mapred-site.xml.template” to “mapred-site.xml” and edit this file C:/Hadoop-3.3.0/etc/hadoop/mapred-site.xml, paste xml code and save this file.**

=====

```
<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>
```

=====

**Create folder “data” under “C:\Hadoop-3.3.0”**

**Create folder “datanode” under “C:\Hadoop-3.3.0\data”**

**Create folder “namenode” under “C:\Hadoop-3.3.0\data”**

=====

**Edit file C:\Hadoop-3.3.0/etc/hadoop/hdfs-site.xml,**

**paste xml code and save this file.**

```
<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>
```

```
<name>dfs.namenode.name.dir</name>
<value>/hadoop-3.3.0/data/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/hadoop-3.3.0/data/datanode</value>
</property>
</configuration>
```

=====

**Edit file C:/Hadoop-3.3.0/etc/hadoop/yarn-site.xml,  
paste xml code and save this file.**

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>127.0.0.1:8032</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>127.0.0.1:8030</value>
</property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>127.0.0.1:8031</value>
```

</property>

</configuration>

=====

## 6. Edit file C:/Hadoop-3.3.0/etc/hadoop/hadoop-env.cmd

Find "JAVA\_HOME=%JAVA\_HOME%" and replace it as

set JAVA\_HOME="C:\Java\jdk1.8.0\_361"

=====

## 7. Download "redistributable" package

Download and run VC\_redist.x64.exe

## 8. Hadoop Configurations

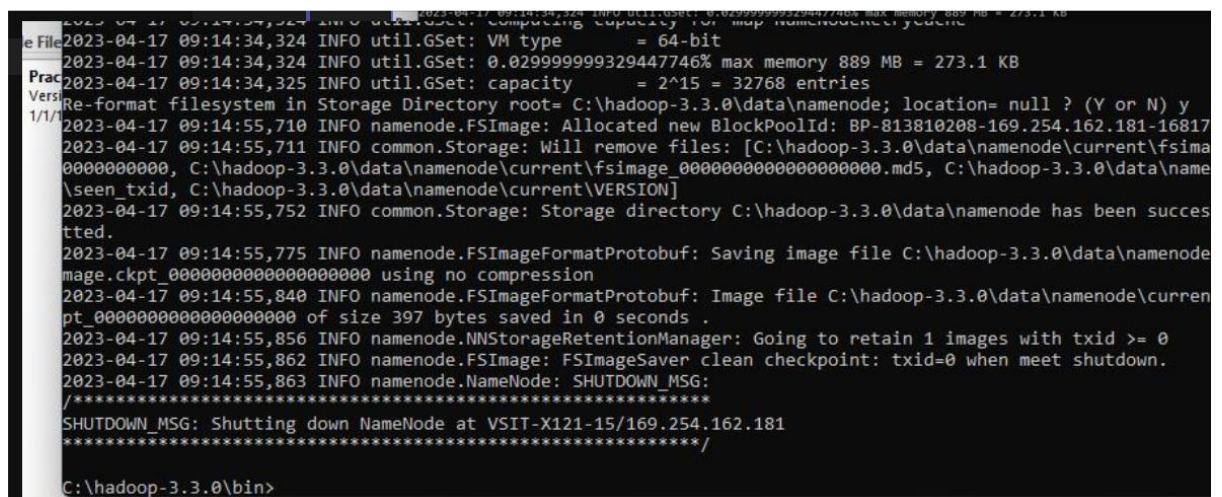
Download bin folder from <https://github.com/s911415/apache-hadoop-3.1.0-winutils>

– Copy the bin folder to c:\hadoop-3.3.0. Replace the existing bin folder.

9. copy "hadoop-yarn-server-timelineservice-3.0.3.jar" from ~\hadoop-3.0.3\share\hadoop\yarn\timelineservice to ~\hadoop-3.0.3\share\hadoop\yarn folder.

## 10. Format the NameNode

– Open cmd 'Run as Administrator' and type command "hdfs namenode –format"



```

2023-04-17 09:14:34,324 INFO util.GSet: VM type = 64-bit
2023-04-17 09:14:34,324 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
2023-04-17 09:14:34,325 INFO util.GSet: capacity = 2^15 = 32768 entries
Re-format filesystem in Storage Directory root= C:\hadoop-3.3.0\data\namenode; location= null ? (Y or N) y
2023-04-17 09:14:55,710 INFO namenode.FSImage: Allocated new BlockPoolId: BP-813810208-169.254.162.181-16817
2023-04-17 09:14:55,711 INFO common.Storage: Will remove files: [C:\hadoop-3.3.0\data\namenode\current\fsima
0000000000, C:\hadoop-3.3.0\data\namenode\current\fsimage_00000000000000000000.md5, C:\hadoop-3.3.0\data\name
\seen_txid, C:\hadoop-3.3.0\data\namenode\current\VERSION]
2023-04-17 09:14:55,752 INFO common.Storage: Storage directory C:\hadoop-3.3.0\data\namenode has been succes
tted.
2023-04-17 09:14:55,775 INFO namenode.FSImageFormatProtobuf: Saving image file C:\hadoop-3.3.0\data\namenode
image.ckpt_00000000000000000000 using no compression
2023-04-17 09:14:55,840 INFO namenode.FSImageFormatProtobuf: Image file C:\hadoop-3.3.0\data\namenode\current
pt_00000000000000000000 of size 397 bytes saved in 0 seconds .
2023-04-17 09:14:55,856 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2023-04-17 09:14:55,862 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2023-04-17 09:14:55,863 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at VSIT-X121-15/169.254.162.181
*****/
C:\hadoop-3.3.0\bin>

```

## 11. Testing

– Open cmd 'Run as Administrator' and change directory to C:\Hadoop-3.3.0\sbin

– type start-all.cmd

OR

- type start-dfs.cmd

- type start-yarn.cmd

```
C:\hadoop-3.3.0\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
The filename, directory name, or volume label syntax is incorrect.
The filename, directory name, or volume label syntax is incorrect.
starting yarn daemons
The filename, directory name, or volume label syntax is incorrect.
```

- You will get 4 more running threads for Datanode, namenode, resource manager and node manager

```

Apache Hadoop Distribution - yarn nodemanager
cbb(HTTP/1.1,[http/1.1]){0.0.0.0:8042}
2023-04-17 09:18:20,128 INFO server.Server: Started @12901ms
2023-04-17 09:18:20,128 INFO webapp.WebApps: Web app node started at 8042
2023-04-17 09:18:20,129 INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is
: VSIT-X121-15:50235
2023-04-17 09:18:20,130 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2023-04-17 09:18:20,135 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting
to ResourceManager at /127.0.0.1:8031
2023-04-17 09:18:20,212 INFO nodemanager.NodeStatusUpdaterImpl: Sending out 0 NM co
ntainer statuses: []
2023-04-17 09:18:20,220 INFO nodemanager.NodeStatusUpdaterImpl: Registering with RM
using containers: []
2023-04-17 09:18:20,408 INFO security.NMContainerTokenSecretManager: Rolling master
-key for container-tokens, got key with id 103481547
2023-04-17 09:18:20,409 INFO security.NMTokenSecretManagerInNM: Rolling master-key
for container-tokens, got key with id 175846895
2023-04-17 09:18:20,409 INFO nodemanager.NodeStatusUpdaterImpl: Registered with Res
ourceManager as VSIT-X121-15:50235 with total resource of <memory:8192, vCores:8>

Administrator: Command Prompt
The filename, directory name, or volume label syntax is incorrect.
The filename, directory name, or volume label syntax is incorrect.
starting yarn daemons
The filename, directory name, or volume label syntax is incorrect.
C:\hadoop-3.3.0\sbin>start-dfs.cmd
The filename, directory name, or volume label syntax is incorrect.
C:\hadoop-3.3.0\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
The filename, directory name, or volume label syntax is incorrect.
The filename, directory name, or volume label syntax is incorrect.
starting yarn daemons
The filename, directory name, or volume label syntax is incorrect.
C:\hadoop-3.3.0\sbin>jps
8164 NameNode
11272 Jps
11576 NodeManager
2616 DataNode

Apache Hadoop Distribution - yarn resourcemanager
hadoop ipc.DefaultRpcScheduler: ipcBackoff: false.
2023-04-17 09:18:20,197 INFO ipc.Server: Starting Socket Reader #1 for port 8032
2023-04-17 09:18:20,200 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.
hadoop.yarn.api.ApplicationClientProtocolPB to the server
2023-04-17 09:18:20,201 INFO ipc.Server: IPC Server Responder: starting
2023-04-17 09:18:20,201 INFO ipc.Server: IPC Server listener on 8032: starting
2023-04-17 09:18:20,390 INFO resourcemanager.ResourceTrackerService: NodeManager fr
om node VSIT-X121-15(cmPort: 50235 httpPort: 8042) registered with capability: <mem
ory:8192, vCores:8>, assigned nodeId VSIT-X121-15:50235
2023-04-17 09:18:20,394 INFO rmnode.RMNodeImpl: VSIT-X121-15:50235 Node Transitione
d from NEW to RUNNING
2023-04-17 09:18:20,420 INFO capacity.CapacityScheduler: Added node VSIT-X121-15:50
235 clusterResource: <memory:8192, vCores:8>
2023-04-17 09:18:20,477 INFO webproxy.ProxyCA: Created Certificate for OU=YARN-d82c
0796-bcc5-4ed3-930e-79d82abbc823
2023-04-17 09:18:20,504 INFO recovery.RMStateStore: Storing CA Certificate and Priv
ate Key
2023-04-17 09:18:20,504 INFO resourcemanager.ResourceManager: Transitioned to activ
e state

Apache Hadoop Distribution - hadoop datanode
-1
2023-04-17 09:18:19,489 INFO datanode.DirectoryScanner: Periodic Directory Tree Ver
ification scan starting in 20168281ms with interval of 21600000ms and throttle limi
t of -1ms/s
2023-04-17 09:18:19,500 INFO datanode.DataNode: Block pool BP-813810208-169.254.162
.181-1681703095701 (Datanode Uuid 0a280d70-04fa-4e1c-8bf0-fa58de4c73cc) service to
localhost/127.0.0.1:9000 beginning handshake with NN
2023-04-17 09:18:19,634 INFO datanode.DataNode: Block pool BP-813810208-169.254.162
.181-1681703095701 (Datanode Uuid 0a280d70-04fa-4e1c-8bf0-fa58de4c73cc) service to
localhost/127.0.0.1:9000 successfully registered with NN
2023-04-17 09:18:19,636 INFO datanode.DataNode: For namenode localhost/127.0.0.1:90
00 using BLOCKREPORT_INTERVAL of 21600000msecs CACHEREPORT_INTERVAL of 10000msecs I
nitial delay: 0msecs; heartBeatInterval=3000
2023-04-17 09:18:19,791 INFO datanode.DataNode: Successfully sent block report 0x1a
ef65b7b2a7b7a4, containing 1 storage report(s), of which we sent 1. The reports ha
d 0 total blocks and used 1 RPC(s). This took 3 msecs to generate and 36 msecs for
RPC and NN processing. Got back one command: FinalizeCommand/5.
2023-04-17 09:18:19,792 INFO datanode.DataNode: Got finalize command for block pool
BP-813810208-169.254.162.181-1681703095701

```

Output:

12. Type JPS command to start-all.cmd command prompt, you will get following output.

```
C:\hadoop-3.3.0\sbin>jps
8164 NameNode
11272 Jps
11576 NodeManager
2616 DataNode
2952 ResourceManager

C:\hadoop-3.3.0\sbin>
```

13. Run <http://localhost:9870/> from any browser



The screenshot shows the Hadoop NameNode Overview page in a web browser. The browser address bar shows the URL `localhost:9870/dfshealth.html#tab-overview`. The page has a green navigation bar with tabs: Hadoop, Overview (selected), Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main content area is titled "Overview 'localhost:9000' (✓active)". Below the title is a table with the following information:

Started:	Mon Apr 17 09:18:18 +0530 2023
Version:	3.3.0, raa96f1871bfd858f9bac59cf2a81ec470da649af
Compiled:	Tue Jul 07 00:14:00 +0530 2020 by brahma from branch-3.3.0
Cluster ID:	CID-7483febe-8254-46e7-bac1-9e1d4d8cbee0
Block Pool ID:	BP-813810208-169.254.162.181-1681703095701

Below the table is a section titled "Summary" with the text "Security is off." at the bottom. The Windows taskbar at the bottom shows the time as 9:21 AM on 4/17/2023.

This is a duplicate of the screenshot above, showing the same Hadoop NameNode Overview page with the same information and layout.



# Practical 2

## Implement word count / frequency programs using MapReduce

**Solution:**

**C:\hadoop-3.3.0\sbin>start-dfs.cmd**

**C:\hadoop-3.3.0\sbin>start-yarn.cmd**

**Open a command prompt as administrator and run the following command to create an input and output folder on the Hadoop file system, to which we will be moving the sample.txt file for our analysis.**

**C:\hadoop-3.3.0\bin>cd\**

**C:\>hadoop dfsadmin -safemode leave**

**DEPRECATED: Use of this script to execute hdfs command is deprecated.**

**Instead use the hdfs command for it.**

**Safe mode is OFF**

**C:\>hadoop fs -mkdir /input\_dir**

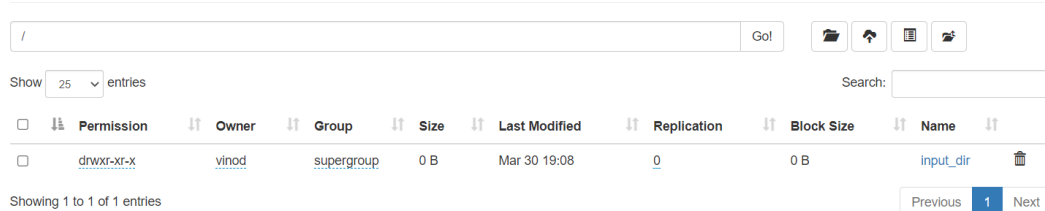
**Check it by giving the following URL at browser**

**http://localhost:9870**

**Utilities -> browse the file system**



### Browse Directory



**Copy the input text file named input\_file.txt in the input directory (input\_dir) of HDFS.**

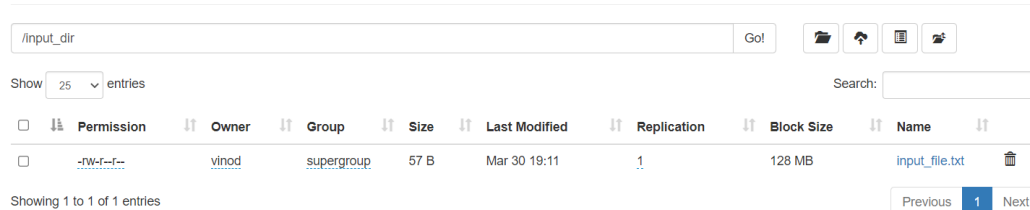
**Make a file in c:\input\_file.txt and write following content in it.**

**Hadoop Window version is easy compared to Ubuntu version**

**Now apply the following command at c:\>**

**C:\> hadoop fs -put C:/input\_file.txt /input\_dir**

### Browse Directory



**Verify input\_file.txt available in HDFS input directory (input\_dir).**

**C:\>Hadoop fs -ls /input\_dir/**

```
C:\>hadoop fs -put C:/input_file.txt /input_dir

C:\>hadoop fs -ls /input_dir/
Found 1 items
-rw-r--r--  1 vinod supergroup          57 2023-03-30 19:11 /input_dir/input_file.txt

C:\>
```

Verify content of the copied file

```
C:\>hadoop dfs -cat /input_dir/input_file.txt
```

You can see the file content displayed on the CMD.

```
C:\>hadoop dfs -cat /input_dir/input_file.txt
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Hadoop Window version is easy compared to Ubuntu version.
C:\>
```

Run MapReduceClient.jar and also provide input and out directories.

```
C:\>hadoop jar C:/hadoop-3.3.0/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.0.jar wordcount /input_dir /output_dir
```

```
Reduce input groups=8
Reduce shuffle bytes=103
Reduce input records=8
Reduce output records=8
Spilled Records=16
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=70
CPU time spent (ms)=219
Physical memory (bytes) snapshot=517128192
Virtual memory (bytes) snapshot=792633344
Total committed heap usage (bytes)=392691712
Peak Map Physical memory (bytes)=314761216
Peak Map Virtual memory (bytes)=465485824
Peak Reduce Physical memory (bytes)=202366976
Peak Reduce Virtual memory (bytes)=327180288
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=56
File Output Format Counters
  Bytes Written=65

:\Windows\System32>
```

In case, there is some error in executing then copy the file MapReduceClient.jar in C:\ and run the program with the jar file using existing MapReduceClient.jar file as:

```
C:\> hadoop jar C:/MapReduceClient.jar wordcount /input_dir /output_dir
```

Now, check the output\_dir on browser as follows:

**Browse Directory**

/ Go! [Icons]

Show 25 entries Search: [ ]

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	vfnod	supergroup	0 B	Mar 30 19:29	0	0 B	input_dir
drwxr-xr-x	vfnod	supergroup	0 B	Mar 30 19:30	0	0 B	output_dir

**Click on output\_dir → part-r-00000 → Head the file (first 32 K) and check the file content as the output.**

File information - part-r-00000

Download Head the file (first 32K) Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073741832  
 Block Pool ID: BP-537931513-192.168.1.19-1680861805234  
 Generation Stamp: 1008  
 Size: 65  
 Availability:  
 • DESKTOP-OL8EULH

File contents

```
Hadoop 1
Ubuntu 1
Window 1
compared 1
easy 1
is 1
to 1
version 2
```

Block Size Name

MB \_SUCCESS

MB part-r-00000

Previous 1 Next

Alternatively, you may type the following command on CMD window as:

```
C:\> hadoop dfs -cat /output_dir/*
```

You can get the following output

```
C:\Windows\System32>hadoop dfs -cat /output_dir/*
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Hadoop 1
Ubuntu 1
Window 1
compared 1
easy 1
is 1
to 1
version 2
C:\Windows\System32>
```

**Output:**



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Hadoop-3.3.0\sbin

C:\hadoop-3.3.0\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\hadoop-3.3.0\sbin>jps
1584 DataNode
11028 Jps
6952 NodeManager
6476 NameNode
9308 ResourceManager

C:\hadoop-3.3.0\sbin>cd\

C:\>hadoop dfsadmin -safemode leave
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Safe mode is OFF

C:\>hadoop fs -mkdir /input_dir

C:\>hadoop fs -put C:/input_file.txt /input_dir
put: `/input_file.txt': No such file or directory

C:\>hadoop fs -put C:/aditi.txt /input_dir

C:\>hadoop jar C:/hadoop-3.3.0/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.0.jar wordcount /input_dir /output_dir
2023-04-17 09:35:32 570 INFO client.DefaultHadoopFilesystemProvider: Connecting to ResourceManager at /127.0.0.1:8032
```

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\Hadoop-3.3.0\sbin

C:\hadoop-3.3.0\sbin>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\hadoop-3.3.0\sbin>jps
1584 DataNode
11028 Jps
6952 NodeManager
6476 NameNode
9308 ResourceManager

C:\hadoop-3.3.0\sbin>cd\

C:\>hadoop dfsadmin -safemode leave
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
Safe mode is OFF

C:\>hadoop fs -mkdir /input_dir

C:\>hadoop fs -put C:/input_file.txt /input_dir
put: `/input_file.txt': No such file or directory

C:\>hadoop fs -put C:/aditi.txt /input_dir

C:\>hadoop jar C:/hadoop-3.3.0/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.0.jar wordcount /input_dir /output_dir
2023-04-17 09:35:32 570 INFO client.DefaultHadoopFilesystemProvider: Connecting to ResourceManager at /127.0.0.1:8032
```

## Practical 3

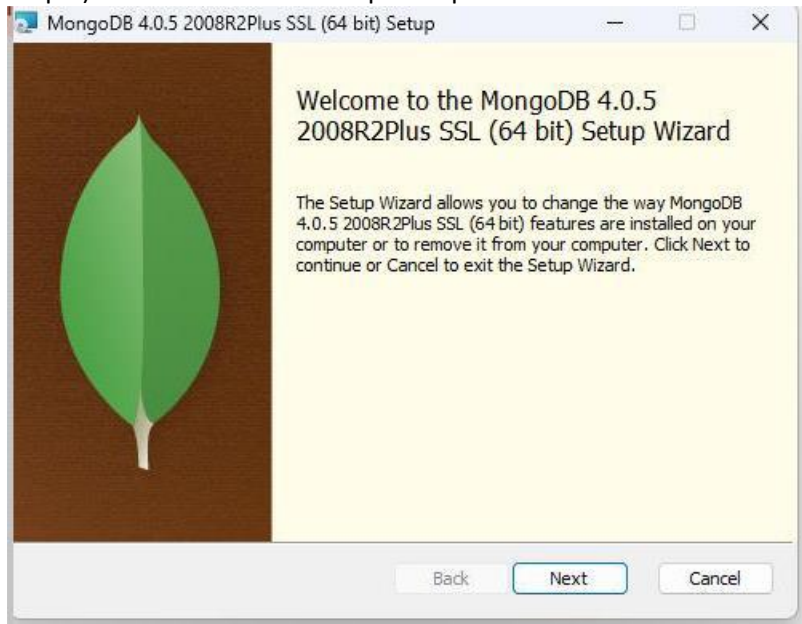
### Implement an application that stores big data in Hbase / MongoDB and manipulate it using R / Python

- a. PyMongo
- b. Mongo Database

#### Step A: Install Mongo database

Step 1) Go to (<https://www.mongodb.com/download-center/community>) and Download MongoDB Community Server. We will install the 64-bit version for Windows.

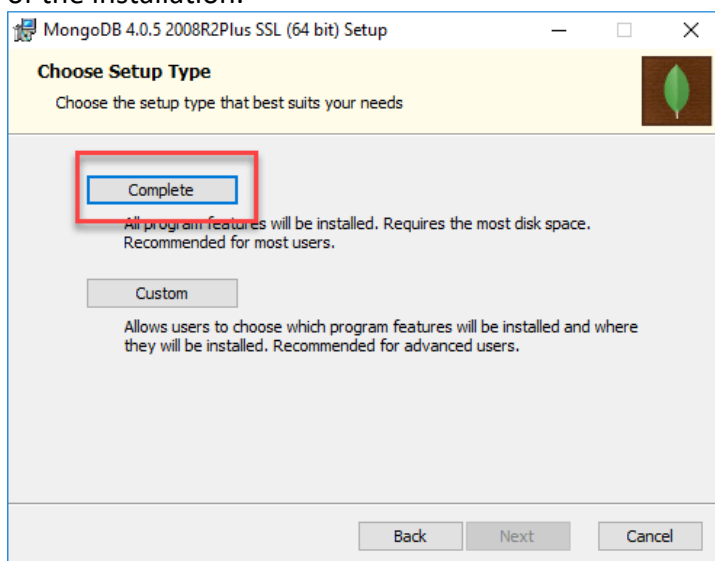
Step 2) Once download is complete open the msi file. Click Next in the start up screen



Step 3)

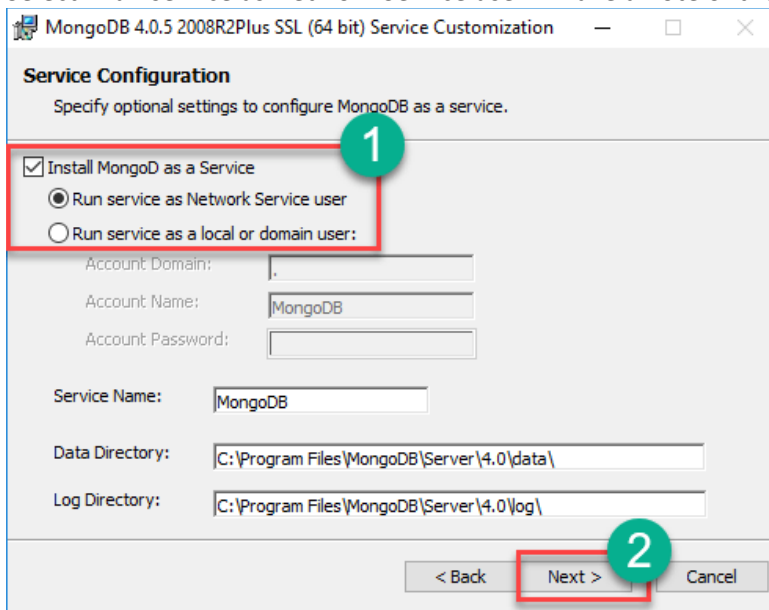
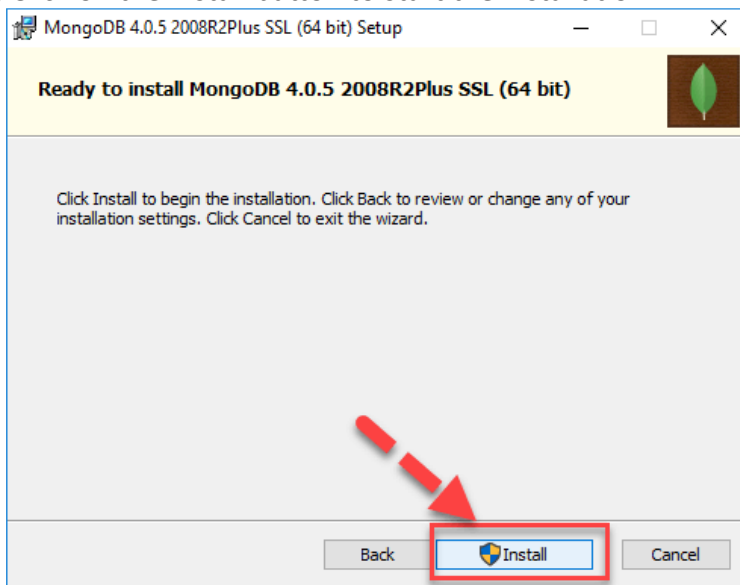
- 1. Accept the End-User License Agreement
- 2. Click Next

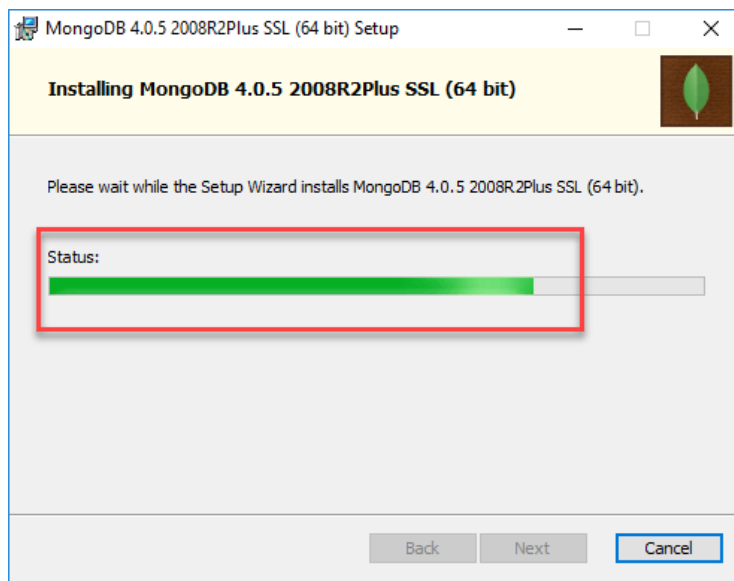
Step 4) Click on the "complete" button to install all of the components. The custom option can be used to install selective components or if you want to change the location of the installation.



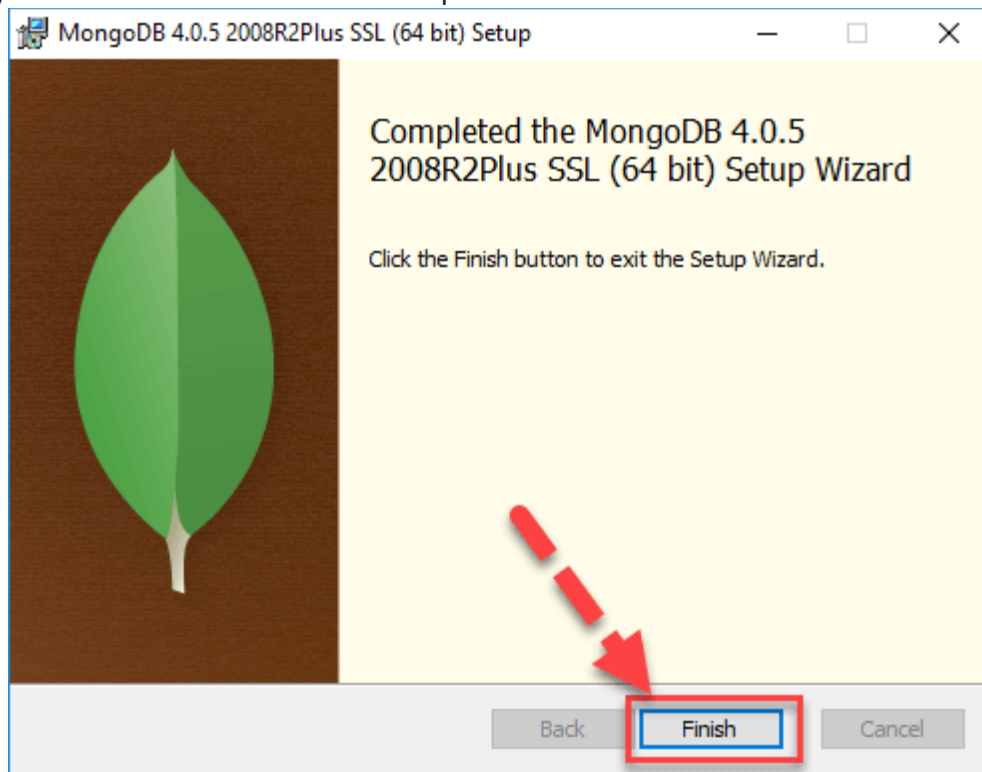
**Step 5)**

1. Select "Run service as Network Service user". make a note of the data directory,

**Step 6)** Click on the Install button to start the installation.**Step 7)** Installation begins. Click Next once completed.



**Step 8)** Click on the Finish button to complete the installation



### Test Mongoddb

**Step 1)** Go to " C:\Program Files\MongoDB\Server\4.0\bin" and double click on **mongo.exe**.

Alternatively, you can also click on the MongoDB desktop icon.

Create the directory where MongoDB will store its files.

Open command prompt window and apply following commands

```
C:\users\admin> cd\
```

```
C:\>md data\db
```

```
2023-04-19T08:03:55.694+0530 I INDEX [LogicalSessionCacheRefresh] build index on: config.system.s
{ v: 2, key: { lastUse: 1 }, name: "lsidTTLIndex", ns: "config.system.sessions", expireAfterSeconds:
2023-04-19T08:03:55.695+0530 I INDEX [LogicalSessionCacheRefresh] building index using bulk m
porarily use up to 500 megabytes of RAM
2023-04-19T08:03:55.699+0530 I INDEX [LogicalSessionCacheRefresh] build index done. scanned 0 to
```



**Step 2) Execute mongod**

Open another command prompt window.

```
C:\> cd C:\Program Files\MongoDB\Server\4.0\bin
```

```
C:\Program Files\MongoDB\Server\4.0\bin> mongod
```

*In case if it gives an error then run the following command:*

```
C:\Program Files\MongoDB\Server\4.0\bin> mongod -repair
```

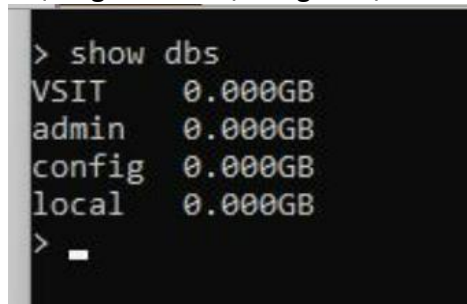
**Step 3) Connect to MongoDB using the Mongo shell**

Let the MongoDB daemon to run.

Open another command prompt window and run the following commands:

```
C:\users\admin> cd C:\Program Files\MongoDB\Server\4.0\bin
```

```
C:\Program Files\MongoDB\Server\4.0\bin> mongo
```



```
> show dbs
VSIT      0.000GB
admin     0.000GB
config    0.000GB
local     0.000GB
> _
```

**Step 4) Install PyMongo**

Open another command prompt window and run the following commands:

Check the python version on your desktop / laptop and copy that path from window explorer

```
C:\users\admin>cd C:\Program Files\Python311\Scripts
```

```
C:\Program Files\Python38\Scripts > python -m pip install pymongo
```

**Step 5) Test PyMongo**

Run the following command from python command prompt

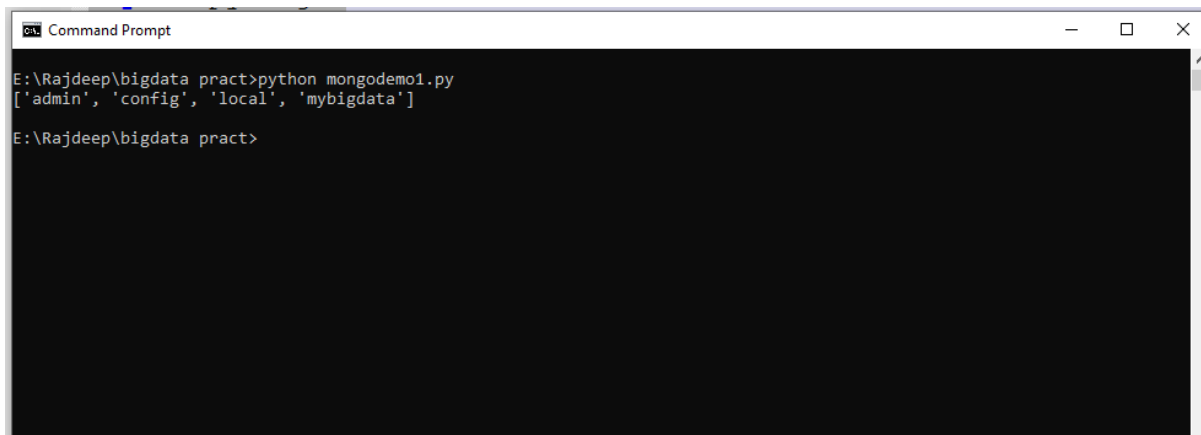
```
import pymongo
```

Now, either create a file in Python IDLE or run all commands one by one in sequence on Python cell

**Program 1:** Displaying the database name:

```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mybigdata"]
print(myclient.list_database_names())
```

Output:

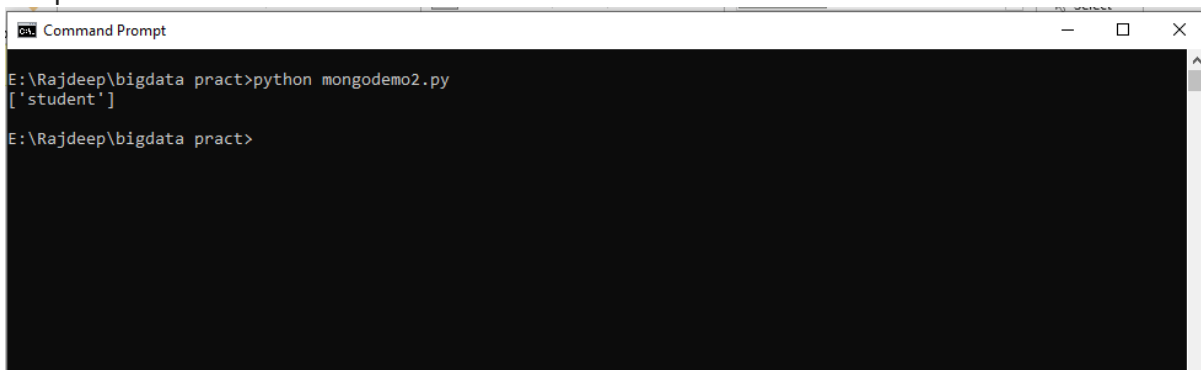


```
Command Prompt
E:\Rajdeep\bigdata pract>python mongodemo1.py
['admin', 'config', 'local', 'mybigdata']
E:\Rajdeep\bigdata pract>
```

**Program 2:** Creating collection:

```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mybigdata"]
mycol=mydb["student"]
print(mydb.list_collection_names())
```

Output:

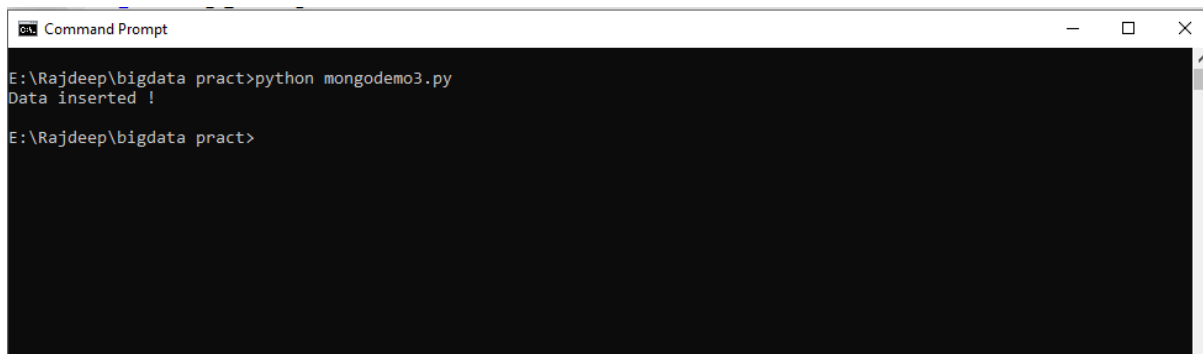


```
Command Prompt
E:\Rajdeep\bigdata pract>python mongodemo2.py
['student']
E:\Rajdeep\bigdata pract>
```

**Program 3:** Inserting Data

```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mybigdata"]
mycol=mydb["student"]
mydict={"name":"vai", "address":"bhy"}
x=mycol.insert_one(mydict)
print("Data inserted !")
```

Output:

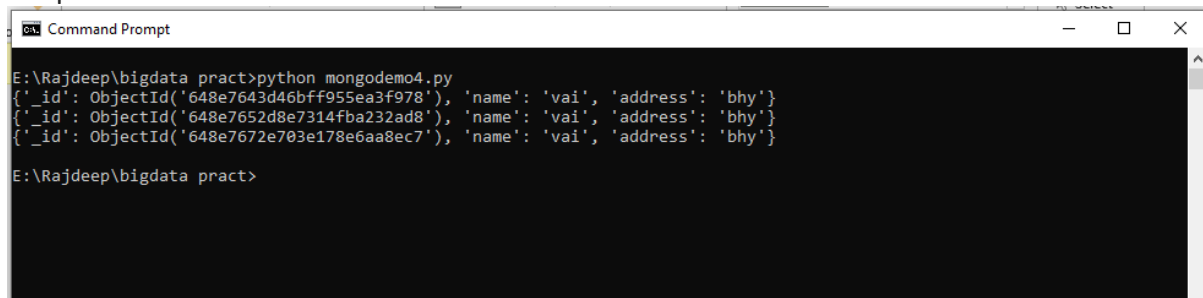


```
Command Prompt
E:\Rajdeep\bigdata pract>python mongodemo3.py
Data inserted !
E:\Rajdeep\bigdata pract>
```

**Program 4:** Insert Multiple data into Collection

```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mybigdata"]
mycol=mydb["student"]
mylist=[{"name":"Ganesh", "address":"Mumbai"}, {"name":"Varun", "address":"Mumbai"},
{"name":"Prasoon", "address":"Pune"}, {"name":"Satish", "address":"Pune"},]
x=mycol.insert_many(mylist)
print("Data inserted !")
```

Output:



```
Command Prompt
E:\Rajdeep\bigdata pract>python mongodemo4.py
{'_id': ObjectId('648e7643d46bff955ea3f978'), 'name': 'vai', 'address': 'bhy'}
{'_id': ObjectId('648e7652d8e7314fba232ad8'), 'name': 'vai', 'address': 'bhy'}
{'_id': ObjectId('648e7672e703e178e6aa8ec7'), 'name': 'vai', 'address': 'bhy'}
E:\Rajdeep\bigdata pract>
```

**Program 5:** Displaying the collection data:

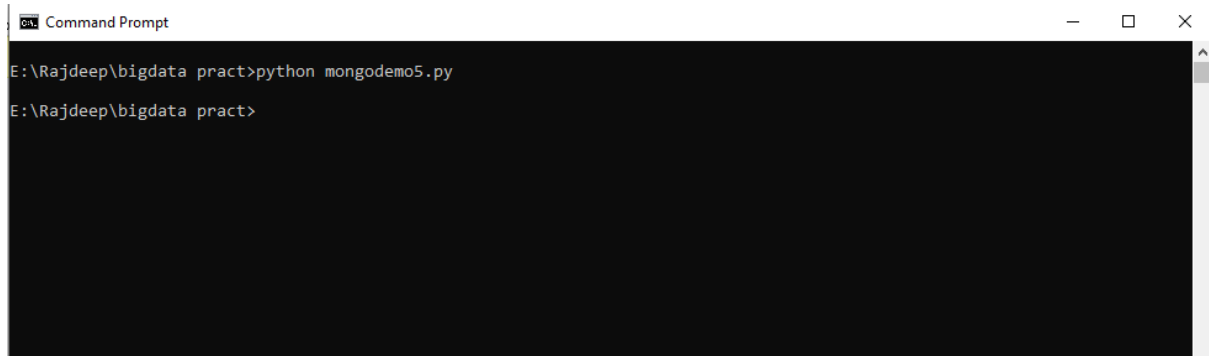
```
import pymongo
myclient = pymongo.MongoClient("mongodb://localhost:27017/")
mydb = myclient["mybigdata"]
mycol = mydb["student"]

myquery = { "name": "Vai" }

mydoc = mycol.find(myquery)

for x in mydoc:
    print(x)
```

Output:



The image shows a screenshot of a Windows Command Prompt window. The title bar at the top reads "Command Prompt". The window has standard Windows window controls (minimize, maximize, close) on the right. The command prompt shows the following text:

```
E:\Rajdeep\bigdata pract>python mongodemo5.py  
E:\Rajdeep\bigdata pract>
```

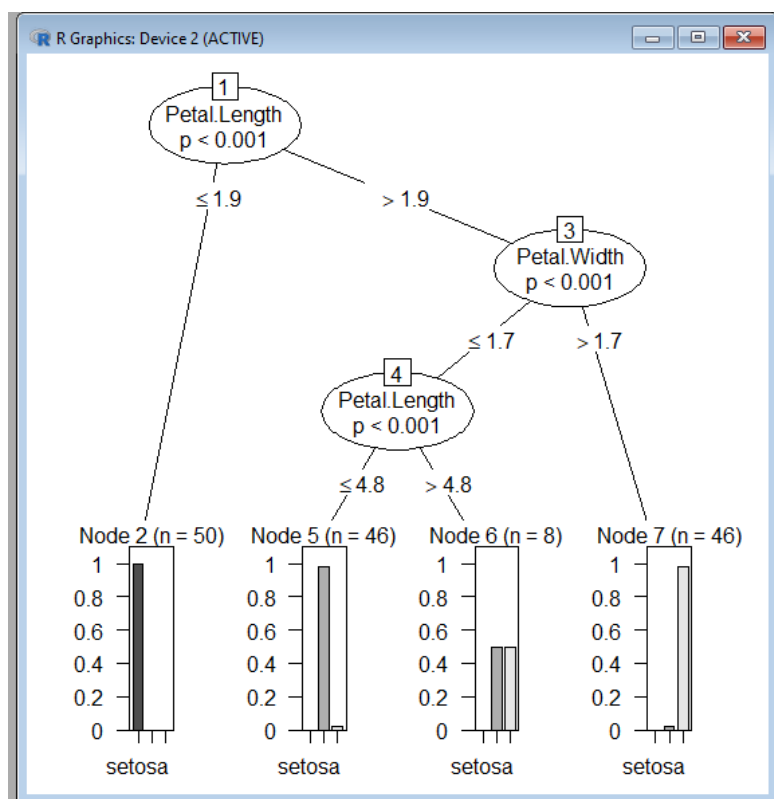
# Practical 4

## Implement Decision tree classification techniques

### Code-

```
library("party")  
print(head(readingSkills))  
str(iris)  
iris_ctree <- ctree(Species ~ Sepal.Width + Sepal.Length + Petal.Length + Petal.Width,  
data=iris)  
print (iris_ctree)  
plot(iris_ctree)
```

### output



# Practical 5

## Implement SVM classification techniques

### Code-

```
# Importing the dataset

dataset = read.csv('E:/NIKHILESH/social.csv')


# Selecting relevant columns: Age, EstimatedSalary, Purchased

dataset = dataset[3:5]


# Encoding the target feature as factor

dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1))


# Splitting the dataset into the Training set and Test set

install.packages('caTools') # Run only once

library(caTools)

set.seed(123)

split = sample.split(dataset$Purchased, SplitRatio = 0.75)

training_set = subset(dataset, split == TRUE)

test_set = subset(dataset, split == FALSE)


# Feature Scaling

training_set[-3] = scale(training_set[-3])

test_set[-3] = scale(test_set[-3])


# Fitting SVM to the Training set

install.packages('e1071') # Run only once

library(e1071)

classifier = svm(formula = Purchased ~ .,
                 data = training_set,
                 type = 'C-classification',
```

```
kernel = 'linear')

# Predicting the Test set results
y_pred = predict(classifier, newdata = test_set[-3])

# Making the Confusion Matrix
cm = table(test_set[, 3], y_pred)
print("Confusion Matrix:")
print(cm)

# Visualising the Training set results
install.packages("ElemStatLearn") # Run only once
library(ElemStatLearn)

# Plotting function (for training and test sets)
plot_svm <- function(set, title) {
  X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
  X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
  grid_set = expand.grid(X1, X2)
  colnames(grid_set) = c('Age', 'EstimatedSalary')
  y_grid = predict(classifier, newdata = grid_set)
  plot(set[, -3],
        main = title,
        xlab = 'Age', ylab = 'Estimated Salary',
        xlim = range(X1), ylim = range(X2))
  contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add = TRUE)
  points(grid_set, pch = '.', col = ifelse(y_grid == 1, 'springgreen3', 'tomato'))
  points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))
}

# Plot training and test results
```



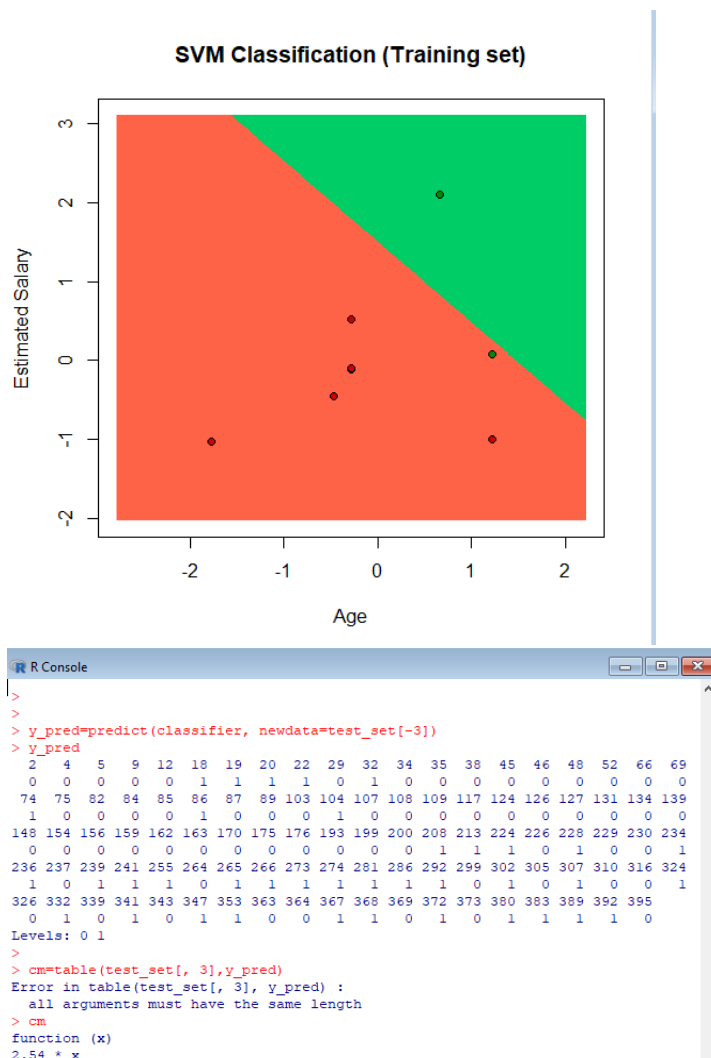
```
plot_svm(training_set, 'SVM Classification (Training set)')
```

```
> set = training_set
> x1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)
> x2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)
```

```
> grid_set = expand.grid(x1, x2)
> colnames(grid_set) = c('Age', 'EstimatedSalary')
> y_grid = predict(classifier, newdata = grid_set)
> plot(set[, -3],
+       main = 'SVM (Training set)',
+       xlab = 'Age', ylab = 'Estimated Salary',
+       xlim = range(x1), ylim = range(x2))
```

```
> contour(X1, X2, matrix(as.numeric(y_grid), length(X1), length(X2)), add = TRUE)
> points(grid_set, pch = '.', col = ifelse(y_grid == 1, 'coral1', 'aquamarine'))
> points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))
```

### Output:



## Practical 6

**REGRESSION MODEL** Import a data from web storage. Name the dataset and now do Logistic Regression to find out relation between variables that are affecting the admission of a student in an institute based on his or her GRE score, GPA obtained and rank of the student. Also check the model is fit or not. **require (foreign), require(MASS).**

Linear regression practical

**code-**

# Load dataset

college <-

read.csv("https://raw.githubusercontent.com/ropensci/datapack/main/inst/extdata/pkg-example/binary.csv")

head(college)

nrow(college)

# Install and load caTools

install.packages("caTools") # Run only once

library(caTools)

# Split dataset

set.seed(123)

split <- sample.split(college\$admit, SplitRatio = 0.75)

training\_reg <- subset(college, split == TRUE)

test\_reg <- subset(college, split == FALSE)

# Fit logistic regression model

fit\_logistic\_model <- glm(admit ~ ., data = training\_reg, family = "binomial")

# View coefficients

coef(fit\_logistic\_model)["gre"]

coef(fit\_logistic\_model)["gpa"]

coef(fit\_logistic\_model)["rank"]

# Predict probabilities on test data

predict\_reg <- predict(fit\_logistic\_model, newdata = test\_reg, type = "response")

# Plot Conditional Density

cdplot(as.factor(admit) ~ gpa, data = college)

cdplot(as.factor(admit) ~ gre, data = college)

cdplot(as.factor(admit) ~ rank, data = college)

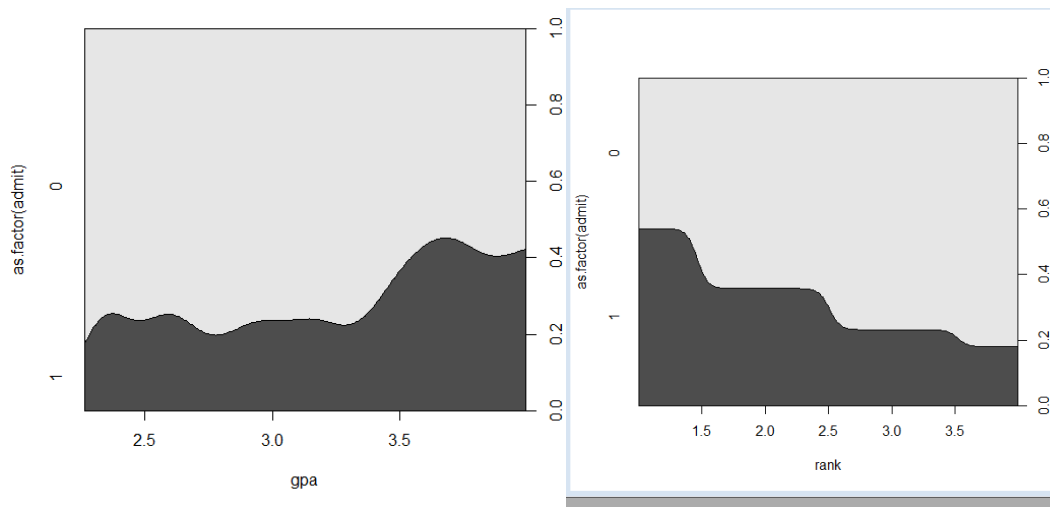
# Convert probabilities to binary predictions

predict\_binary <- ifelse(predict\_reg > 0.5, 1, 0)

# Confusion Matrix

```
table(Actual = test_reg$admit, Predicted = predict_binary)
```

**output**



# Practical 7

**MULTIPLE REGRESSION MODEL:** Apply multiple regressions, if data have a continuous independent variable. Apply on above dataset.

## Code-

Explain Multiple regression in detail.

# Load dataset

```
college <- read.csv("https://raw.githubusercontent.com/csquared/udacity-dlnd/master/nn/binary.csv")
```

```
head(college)
```

```
nrow(college)
```

# Install and load caTools (only run install once)

```
install.packages("caTools") # Only the first time
```

```
library(caTools)
```

# Data splitting

```
set.seed(123)
```

```
split <- sample.split(college$admit, SplitRatio = 0.75)
```

```
training_reg <- subset(college, split == TRUE)
```

```
test_reg <- subset(college, split == FALSE)
```

# Fit logistic regression model

```
fit_MRegressor_model <- glm(formula = admit ~ gre + gpa + rank, data = training_reg, family = binomial)
```

# Predict probabilities on test set

```
predict_reg <- predict(fit_MRegressor_model, newdata = test_reg, type = "response")
```

```
head(predict_reg)
```

# Classify predictions (threshold = 0.5)

```
predict_class <- ifelse(predict_reg > 0.5, 1, 0)
```

**# Plot conditional density plots**

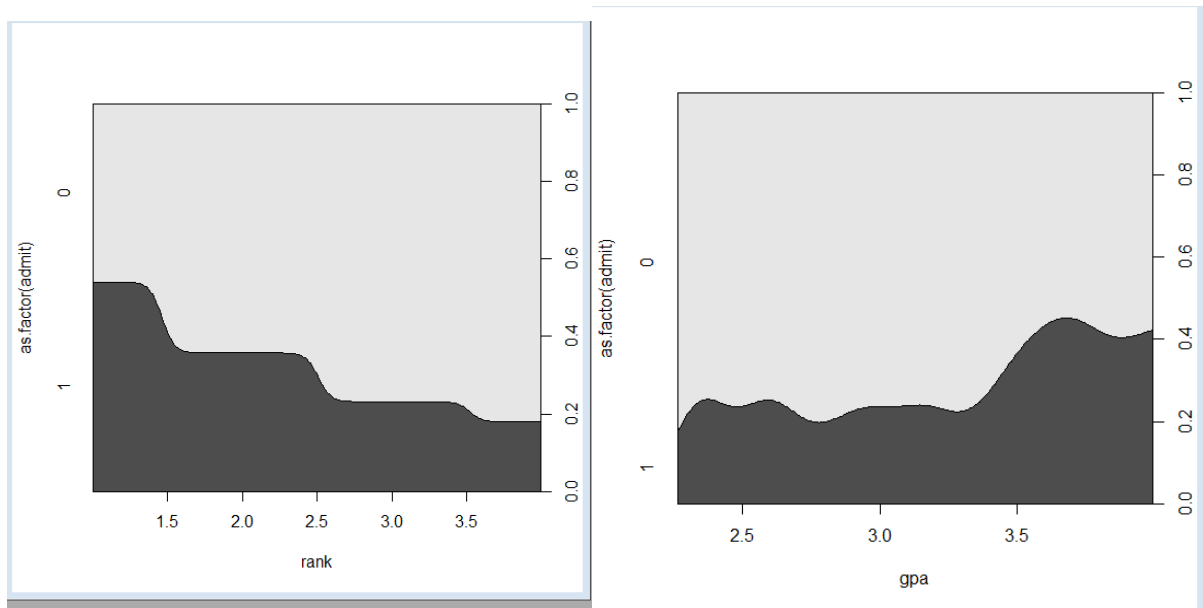
```
cdplot(as.factor(admit) ~ gpa, data = college)
```

```
cdplot(as.factor(admit) ~ gre, data = college)
```

```
cdplot(as.factor(admit) ~ rank, data = college)
```

**# Confusion matrix**

```
table(Actual = test_reg$admit, Predicted = predict_class)
```

**Output**

## Practical 8

CLASSIFICATION MODEL a. Install relevant package for classification. b. Choose classifier for classification problem. c. Evaluate the performance of classifier.

### **Navebyse**

```
data(iris)
str(iris)
install.packages("e1071")
install.packages("caTools")
install.packages("caret")
library(e1071)
library(caTools)
library(caret)
split <- sample.split(iris,SplitRatio=0.7)
train_c1 <-subset(iris,split=="TRUE")
test_c1 <- subset(iris,split == "FALSE")
train_scale <- scale(train_c1[, 1:4])
test_scale <- scale(test_c1[,1:4])

set.seed(120)
classifier_c1 <- naiveBayes(Species ~ ., data = train_c1)
classifier_c1

y_pred <- predict(classifier_c1, newdata= test_c1)
cm <- table(test_c1$Species, y_pred)
cm
confusionMatrix(cm) data(iris)
str(iris)
install.packages("e1071")
install.packages("caTools")
install.packages("caret")
library(e1071)
library(caTools)
library(caret)
split <- sample.split(iris,SplitRatio=0.7)
train_c1 <-subset(iris,split=="TRUE")
test_c1 <- subset(iris,split == "FALSE")
train_scale <- scale(train_c1[, 1:4])
test_scale <- scale(test_c1[,1:4])

set.seed(120)
classifier_c1 <- naiveBayes(Species ~ ., data = train_c1)
classifier_c1
output -
```

```

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
  setosa versicolor virginica
0.3333333 0.3333333 0.3333333

Conditional probabilities:
  Sepal.Length
Y      [,1]      [,2]
setosa  5.050000 0.3739353
versicolor 5.833333 0.5141671
virginica  6.626667 0.5545631

  Sepal.Width
Y      [,1]      [,2]
setosa  3.443333 0.3838852
versicolor 2.753333 0.3191944
virginica  2.976667 0.3058998

```

```
y_pred <- predict(classifier_c1, newdata= test_c1)
```

```
cm <- table(test_c1$Species, y_pred)
```

```
cm
```

```
confusionMatrix(cm)
```

**output-**

```

cm
      y_pred
      setosa versicolor virginica
setosa      20          0          0
versicolor   0         18          2
virginica    0          1         19
confusionMatrix(cm)

```

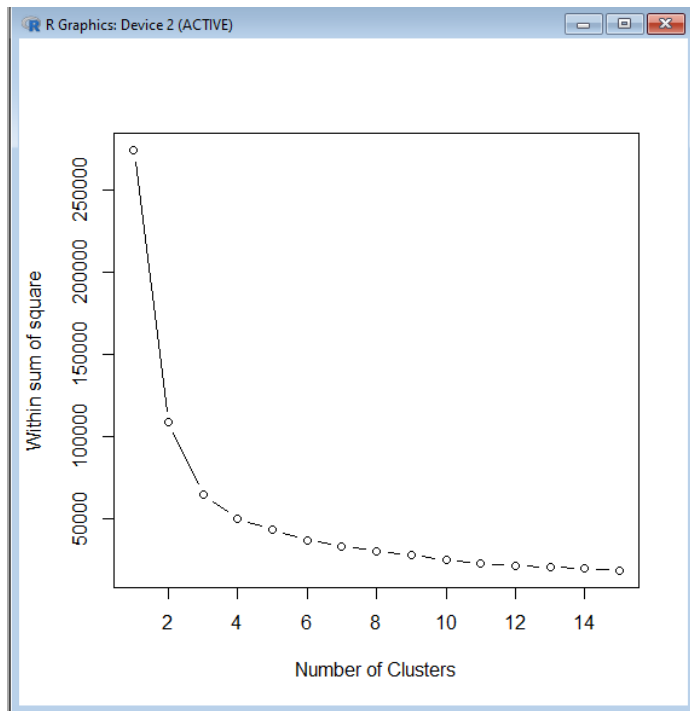


## Practical 9

### CLUSTERING MODEL a. Clustering algorithms for unsupervised classification.

#### b. Plot the cluster data using R visualizations.

```
install.packages("plyr")
install.packages("ggplot2")
install.packages("cluster")
install.packages("lattice")
install.packages("grid")
install.packages("gridExtra")
library(plyr)
library(ggplot2)
library(cluster)
library(lattice)
library(grid)
library(gridExtra)
grade_input=as.data.frame(read.csv("E:/Rajdeep/bigdata
pract/dataset/grades_km_input.csv"))
kmdata_orig=as.matrix(grade_input[, c ("Student","English","Math","Science")])
kmdata=kmdata_orig[,2:4]
kmdata[1:10,]
wss=numeric(15)
for(k in 1:15)wss[k]=sum(kmeans(kmdata,centers=k,nstart=25)$withinss)
plot(1:15,wss,type="b",xlab="Number of Clusters",ylab="Within sum of square")
km = kmeans(kmdata,3,nstart=25)
km
c( wss[3] , sum(km$withinss))
df=as.data.frame(kmdata_orig[,2:4])
df$cluster=factor(km$cluster)
centers=as.data.frame(km$centers)
g1=ggplot(data=df, aes(x=English, y=Math, color=cluster )) +
geom_point() + theme(legend.position="right") +
geom_point(data=centers,aes(x=English,y=Math, color=as.factor(c(1,2,3))),size=10,
alpha=.3, show.legend =FALSE)
g2=ggplot(data=df, aes(x=English, y=Science, color=cluster )) +
geom_point () +geom_point(data=centers,aes(x=English,y=Science,
color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend=FALSE)
g3 = ggplot(data=df, aes(x=Math, y=Science, color=cluster )) +
geom_point () + geom_point(data=centers,aes(x=Math,y=Science,
color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend=FALSE)
tmp=ggplot_gtable(ggplot_build(g1))
grid.arrange(arrangeGrob(g1 + theme(legend.position="none"),g2 +
theme(legend.position="none"),g3 + theme(legend.position="none"),top ="High School
Student Cluster Analysis" ,ncol=1))
```



Aprori Practical

code-

```
library(arules)
```

```
library(arulesViz)
```

```
library(RColorBrewer)
```

```
data(Groceries)
```

```
Groceries
```

```
summary(Groceries)
```

```
class(Groceries)
```

```
rules = apriori(Groceries, parameter = list(supp = 0.02, conf = 0.2))
```

```
summary(rules)
```

```
inspect(rules[1:10])
```

```
arules::itemFrequencyPlot(Groceries, topN = 20,
```

```
col = brewer.pal(8, 'Pastel2'),
```

```
main = 'Relative Item Frequency Plot',
```

```
type = "relative",
```

```
ylab = "Item Frequency (Relative)")
```

```
itemsets = apriori(Groceries, parameter = list(minlen=2, maxlen=2,support=0.02,
```

```
target="frequent itemsets"))
```

```
summary(itemsets)
```

```
inspect(itemsets)
```

```
itemsets_3 = apriori(Groceries, parameter = list(minlen=3, maxlen=3,support=0.02,  
target="frequent itemsets"))  
summary(itemsets_3)
```

```
inspect(itemsets_3)
```

**output-**

