# SRM VALLIAMMAI ENGINEERING COLLEGE

**(An Autonomous Institution)**

SRM Nagar, Kattankulathur-603203

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATASCIENCE**

ACADEMIC YEAR: 2024-2025

EVEN SEMESTER

**LAB MANUAL**

**(REGULATION - 2019)**

# 1922609– BIG DATA ANALYTICS LABORATORY

SIXTH SEMSTER

B.Tech – Artificial Intelligence and Data Science

**Prepared By**

Ms. R. LAKSHMI , A.P (Sel.G) / AI&DS

# INDEX

## PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. To afford the necessary background in the field of Information Technology to deal with engineering problems to excel as engineering professionals in industries.
2. To improve the qualities like creativity, leadership, teamwork and skill thus contributing towards the growth and development of society.
3. To develop ability among students towards innovation and entrepreneurship that caters to the needs of Industry and society.
4. To inculcate and attitude for life-long learning process through the use of information technology sources.
5. To prepare then to be innovative and ethical leaders, both in their chosen profession and in other activities.

## PROGRAMME OUTCOMES (POs)

After going through the four years of study, Information Technology Graduates will exhibit ability to:

| PO# | Graduate Attribute | Programme Outcome |
|---|---|---|
| 1 | Engineering knowledge | Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization for the solution of complex engineering problems. |
| 2 | Problem analysis | Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| 3 | 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 public health and safety, and cultural, societal, and environmental considerations. |
| 4 | 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 |

| | | |
|---|---|---|
| 5 | 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. |
| 6 | 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 |
| 7 | 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. |
| 8 | Ethics | Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice |
| 9 | Individual and team work | Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings |
| 10 | Communication | Communicate effectively on complex engineering activities with the engineering community and with the 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 |
| 11 | 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 |
| 12 | 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 |

## PROGRAMME SPECIFIC OUTCOMES (PSOs)

After the completion of Bachelor of Technology in Artificial Intelligence and Data Science programme the student will have following Program specific outcomes

1. Design and develop secured database applications with data analytical approaches of data preprocessing, optimization, visualization techniques and maintenance using state of the art methodologies based on ethical values.
2. Design and develop intelligent systems using computational principles, methods and systems for extracting knowledge from data to solve real time problems using advanced technologies and tools.
3. Design, plan and setting up the network that is helpful for contemporary business environments using latest software and hardware.
4. Planning and defining test activities by preparing test cases that can predict and correct errors ensuring a socially transformed product catering all technological needs.

**1922609 BIG DATA ANALYTICS LABORATORY L T P C**

**0 0 4 2**

**OBJECTIVES:**

- To optimize business decisions and create competitive advantage with Big Data Analytics.
- To implement Map Reduce programs for processing big data
- To realize storage of big data using H base, Mongo DB
- To analyze big data using linear models
- To analyze big data using machine learning techniques such as SVM / Decisiontree classification and clustering

**LIST OF EXPERIMENTS:**

**Hadoop**

1. Install, configure and run Hadoop and HDFS
2. Implement word count / frequency programs using MapReduce
3. Implement an MR program that processes a weather dataset

**R**

4. Implement Linear and logistic Regression
5. Implement SVM / Decision tree classification techniques
6. Implement clustering techniques
7. Visualize data using any plotting framework
8. Implement an application that stores big data in Hbase / MongoDB / Pig using Hadoop / R

**Total: 60 Periods**

## LIST OF EQUIPMENTS FOR A BATCH OF 30 STUDENTS

**SOFTWARE:**

Hadoop, YARN, R Package, Hbase, MongoDB

**HARDWARE:**

Standalone desktops - 30 Nos. (or) Server supporting 30 terminals or more

**COURSE OUTCOMES**

| | |
|---|---|
| 1922609.1 | Understand and process big data using Hadoop Framework |
| 1922609.2 | Understand and apply regression models |
| 1922609.3 | Understand and perform data analysis with Machine Learning method. |
| 1922609.4 | Analyze and perform graphical data analysis |
| 1922609.5 | Implement and apply tools and techniques to analyses Big data. |

**CO- PO-PSO MATRIX**

| CO | PO | | | | | | | | | | | | PSO | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 1 | 2 | 3 | 4 |
| **1** | 3 | 2 | 1 | - | | - | - | - | 1 | - | - | 3 | - | 2 | - | - |
| **2** | 2 | 2 | - | - | 2 | - | - | - | 1 | - | - | 2 | - | 2 | - | - |
| **3** | 2 | 2 | - | - | | - | - | - | 1 | - | - | 2 | - | 2 | - | - |
| **4** | 3 | 2 | 2 | - | 2 | - | - | - | 1 | - | - | 2 | - | - | - | 2 |
| **5** | 3 | 2 | 1 | - | 2 | - | - | - | - | - | - | 2 | - | - | - | 2 |

## EVALUATION PROCEDURE FOR EACH EXPERIMENT

| S.No | Description | Mark |
|---|---|---|
| 1. | Aim & Pre-Lab discussion | 20 |
| 2. | Observation | 30 |
| 3. | Conduction and Execution | 30 |
| 4. | Output & Result | 10 |
| 5. | Viva | 10 |
| **Total** | | **100** |

## INTERNAL ASSESSMENT FOR LABORATORY

| S.No | Description | Mark |
|---|---|---|
| 1. | Conduction & Execution of Experiment | 50 |
| 2. | Record | 20 |
| 3. | Model Test | 30 |
| **Total** | | **100** |

## Ex.No.1 Install, configure and run Hadoop and HDFS

**AIM**

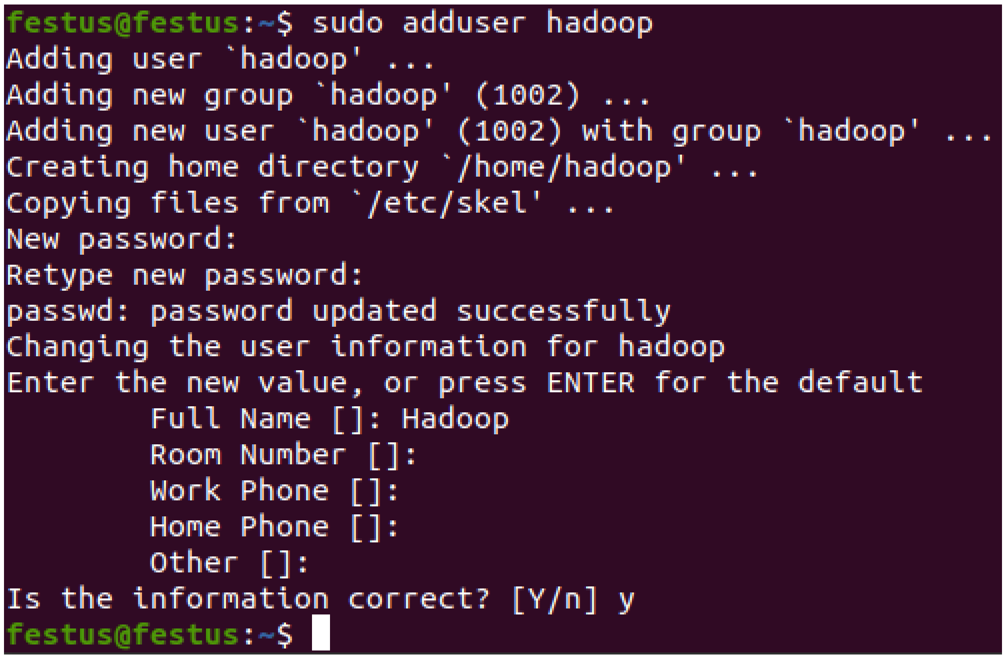
To Study how to install, configure and run Hadoop and HDFS on Ubuntu 20.04.

**PROCEDURE**

**Step by Step Installing Hadoop on Ubuntu 20.04**

**Step 1 — Create user for Hadoop environment**

sudo adduser Hadoop



```
festus@festus:~$ sudo adduser hadoop
Adding user `hadoop' ...
Adding new group `hadoop' (1002) ...
Adding new user `hadoop' (1002) with group `hadoop' ...
Creating home directory `/home/hadoop' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hadoop
Enter the new value, or press ENTER for the default
        Full Name []: Hadoop
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
festus@festus:~$
```

**Step 2— Installing Java**

The following command to update your system before initiating a new installation:

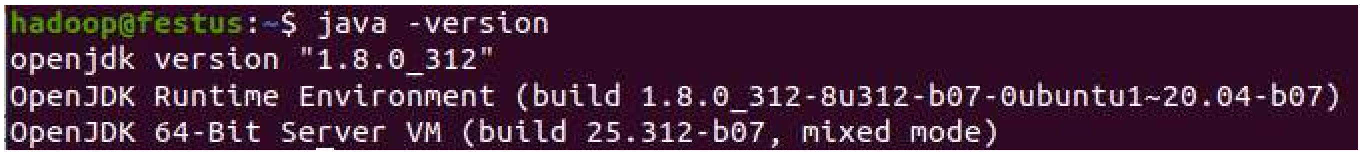
sudo apt update

Install the latest version of Java.

sudo apt install openjdk-8-jdk -y

Once installed, verify the installed version of Java with the following command:

java -version



```
hadoop@festus:~$ java -version
openjdk version "1.8.0_312"
OpenJDK Runtime Environment (build 1.8.0_312-8u312-b07-0ubuntu1~20.04-b07)
OpenJDK 64-Bit Server VM (build 25.312-b07, mixed mode)
```

**Step 3: Install OpenSSH on Ubuntu**

Install the OpenSSH server and client using the following command:

sudo apt install openssh-server openssh-client -y

Switch to the created user.

sudo su - hadoop

Generate public and private key pairs.

$ ssh-keygen -t rsa

Add the generated public key from id_rsa.pub to authorized_keys.

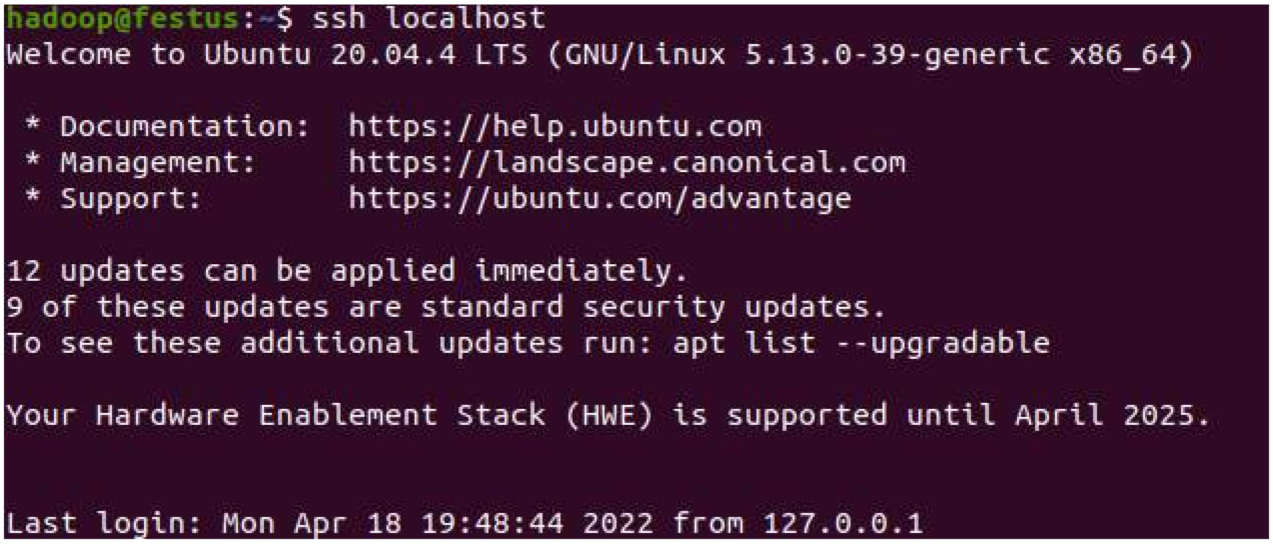
$ sudo cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

Change the permissions of the authorized_keys file.

$ sudo chmod 640 ~/.ssh/authorized_keys

Verify if the password-less SSH is functional.

$ ssh localhost



```
hadoop@festus:~$ ssh localhost
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.13.0-39-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

12 updates can be applied immediately.
9 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.


Last login: Mon Apr 18 19:48:44 2022 from 127.0.0.1
```

**Step 4: Install Apache Hadoop**

Download the latest stable version of Hadoop.

$ wget https://downloads.apache.org/hadoop/common/hadoop-3.3.2/hadoop-3.3.2.tar.gz

Extract the downloaded file.

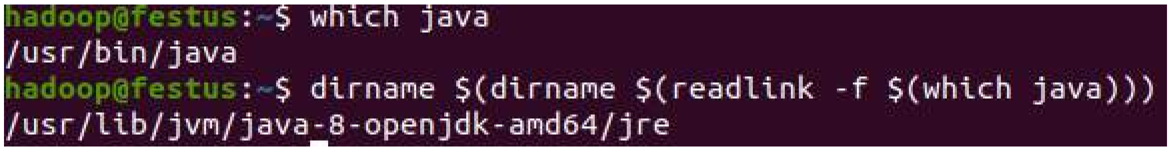
$ tar -xvzf hadoop-3.3.2.tar.gz

Rename the extracted directory as we will do by executing the below-given command:

mv hadoop-3.3.0 hadoop

Now, configure Java environment variables for setting up Hadoop. For this, we will check out the location of our “JAVA_HOME” variable:

dirname $(dirname $(readlink -f $(which java)))



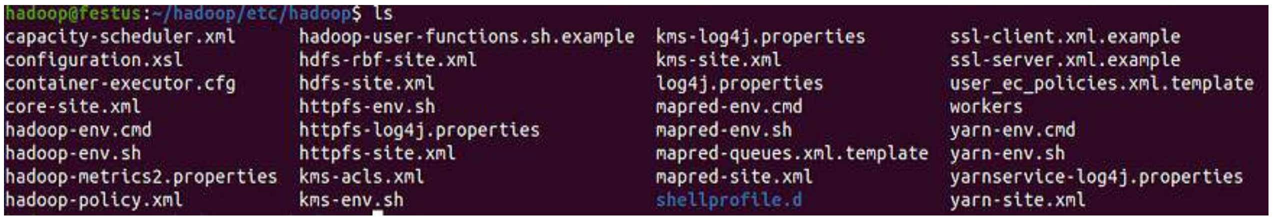
```
hadoop@festus:~$ which java
/usr/bin/java
hadoop@festus:~$ dirname $(dirname $(readlink -f $(which java)))
/usr/lib/jvm/java-8-openjdk-amd64/jre
```

**Step 5: Configure Hadoop**

A Hadoop environment is configured by editing a set of  configuration files:

bashrc, hadoop-env.sh, core-site.xml, hdfs-site.xml, mapred-site-xml and yarn-site.xml

They can be found in the newly created hadoop folder



```
hadoop@festus:~/hadoop/etc/hadoop$ ls
capacity-scheduler.xml       hadoop-user-functions.sh.example  kms-log4j.properties         ssl-client.xml.example
configuration.xsl            hdfs-rbf-site.xml                 kms-site.xml                 ssl-server.xml.example
container-executor.cfg       hdfs-site.xml                     log4j.properties             user_ec_policies.xml.template
core-site.xml                httpfs-env.sh                     mapred-env.cmd               workers
hadoop-env.cmd               httpfs-log4j.properties           mapred-env.sh                yarn-env.cmd
hadoop-env.sh                httpfs-site.xml                   mapred-queues.xml.template   yarn-env.sh
hadoop-metrics2.properties   kms-acls.xml                      mapred-site.xml              yarnservice-log4j.properties
hadoop-policy.xml            kms-env.sh                        shellprofile.d               yarn-site.xml
```

**Step 5a: Configure Hadoop Environment Variables (bashrc)**

Edit file ~/.bashrc to configure the Hadoop environment variables.

$ sudo nano ~/.bashrc

Add the following lines to the file. Save and close the file.

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export HADOOP_HOME=/usr/local/hadoop

export HADOOP_INSTALL=$HADOOP_HOME

export HADOOP_MAPRED_HOME=$HADOOP_HOME

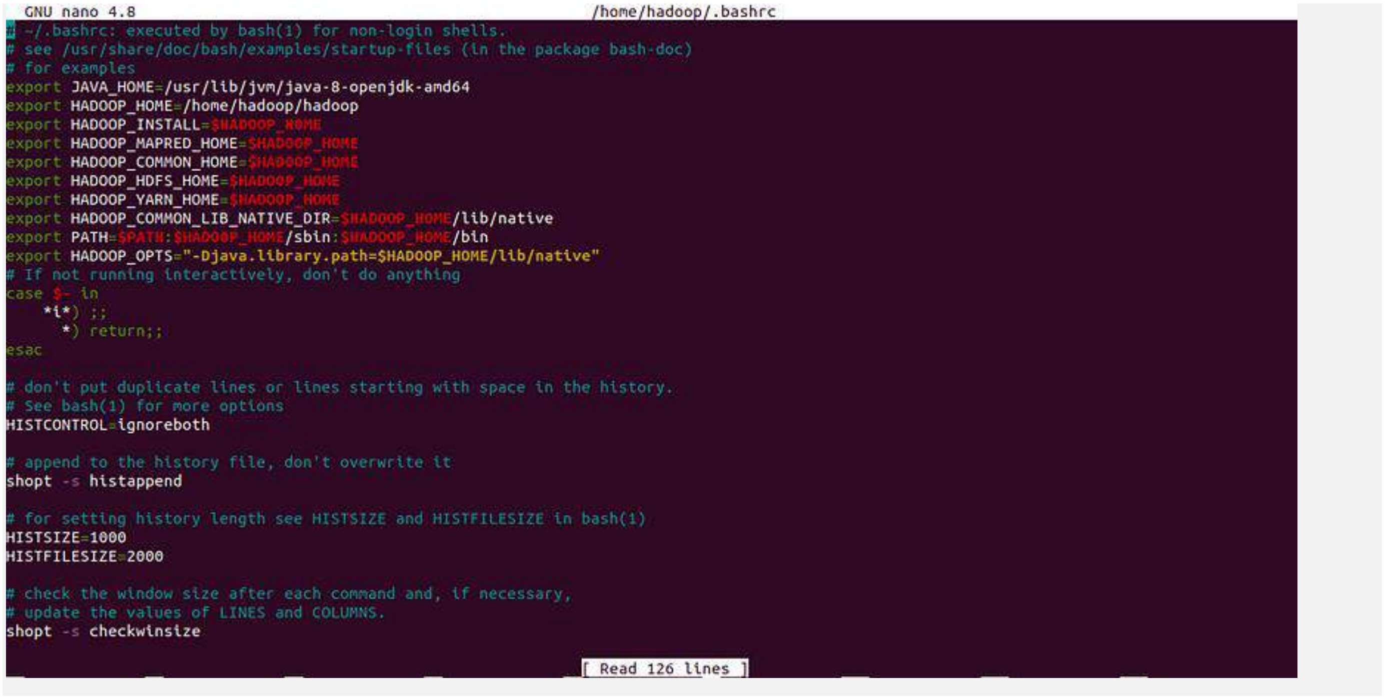
export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME

export YARN_HOME=$HADOOP_HOME

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native

export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"



```
  GNU nano 4.8                                    /home/hadoop/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/hadoop/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

                                  [ Read 126 lines ]
```

Activate the environment variables.

$ source ~/.bashrc

Step 5b: Edit hadoop-env.sh File

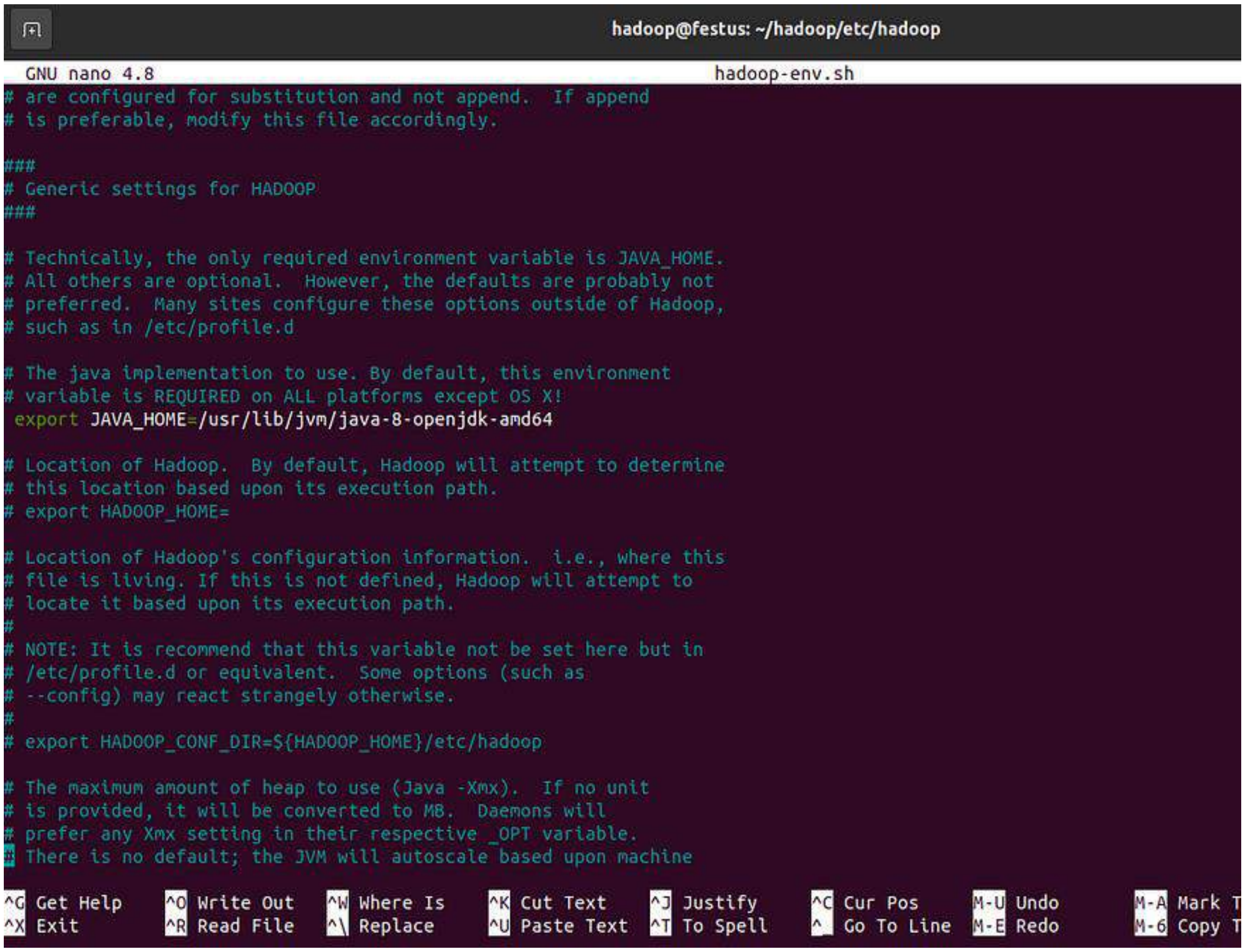
The hadoop-env.sh file serves as a master file to configure YARN, HDFS, MapReduce, and Hadoop-related project settings. When setting up a single node Hadoop cluster, you need to define which Java implementation is to be utilized. Use the previously created $HADOOP_HOME variable to access the hadoop-env.sh file:

sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh

Uncomment the $JAVA_HOME variable (i.e., remove the # sign) and add the full path to the OpenJDK installation on your system.

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

The path needs to match the location of the Java installation on your system.

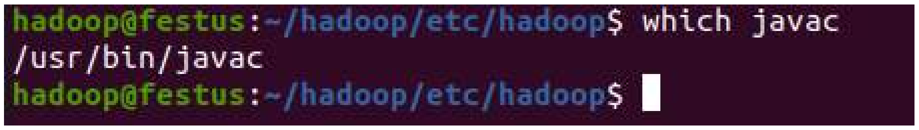


```
hadoop@festus: ~/hadoop/etc/hadoop
  GNU nano 4.8                                  hadoop-env.sh
# are configured for substitution and not append.  If append
# is preferable, modify this file accordingly.

###
# Generic settings for HADOOP
###

# Technically, the only required environment variable is JAVA_HOME.
# All others are optional.  However, the defaults are probably not
# preferred.  Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
 export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

# Location of Hadoop.  By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=

# Location of Hadoop's configuration information.  i.e., where this
# file is living. If this is not defined, Hadoop will attempt to
# locate it based upon its execution path.
#
# NOTE: It is recommend that this variable not be set here but in
# /etc/profile.d or equivalent.  Some options (such as
# --config) may react strangely otherwise.
#
# export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop

# The maximum amount of heap to use (Java -Xmx).  If no unit
# is provided, it will be converted to MB.  Daemons will
# prefer any Xmx setting in their respective _OPT variable.
# There is no default; the JVM will autoscale based upon machine

^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo     M-A Mark T
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line M-E Redo     M-6 Copy T
```

To locate the correct Java path, run the following command in your terminal window:

which javac

The resulting output provides the path to the Java binary directory.



```
hadoop@festus:~/hadoop/etc/hadoop$ which javac
/usr/bin/javac
hadoop@festus:~/hadoop/etc/hadoop$ 
```

Use the provided path to find the OpenJDK directory with the following command:

readlink -f /usr/bin/javac

The section of the path just before the /bin/javac directory needs to be assigned to the $JAVA_HOME variable.

Step 5c: Edit core-site.xml File

The core-site.xml file defines HDFS and Hadoop core properties.

To set up Hadoop in a pseudo-distributed mode, you need to specify the URL for your NameNode, and the temporary directory Hadoop uses for the map and reduce process.

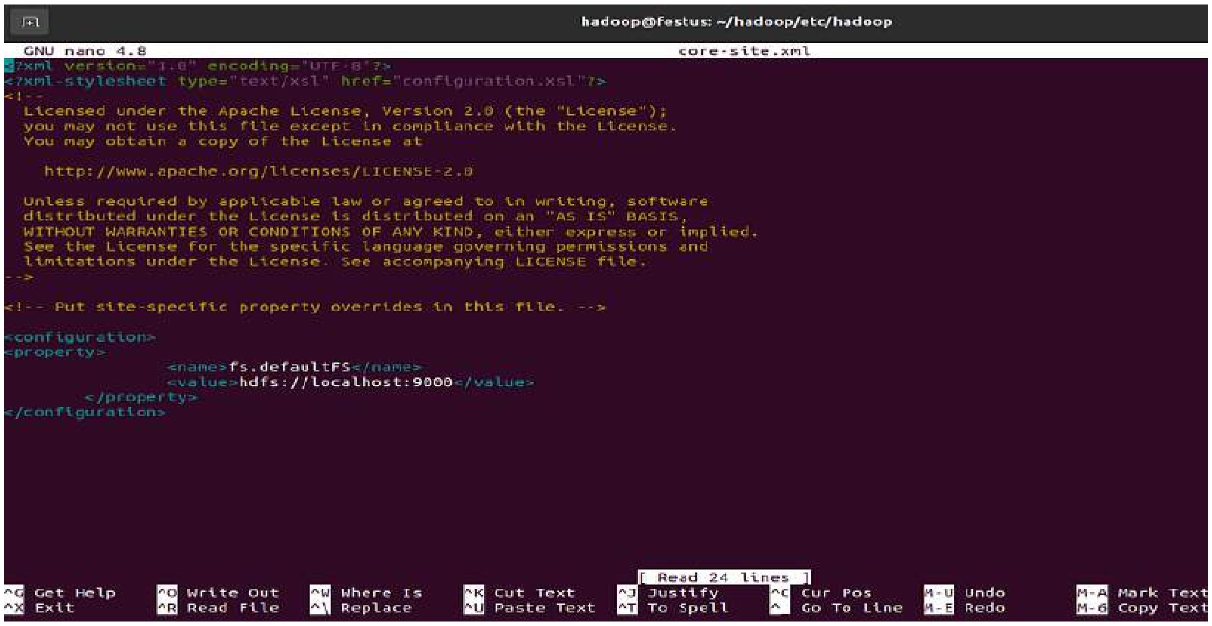
Open the core-site.xml file in a text editor:

```
sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

Add the following configuration to override the default values for the temporary directory and add your HDFS URL to replace the default local file system setting:

```
<configuration>
 <property>
                <name>fs.defaultFS</name>
                <value>hdfs://localhost:9000</value>
        </property>
 </configuration>
```

This example uses values specific to the local system. The data needs to be consistent throughout the configuration process.



```
hadoop@festus: ~/hadoop/etc/hadoop
  GNU nano 4.8                                        core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
                <name>fs.defaultFS</name>
                <value>hdfs://localhost:9000</value>
        </property>
</configuration>

                                          [ Read 24 lines ]
```

**Step 5d: Edit hdfs-site.xml File**

The properties in the hdfs-site.xml file govern the location for storing node metadata, fsimage file, and edit log file. Configure the file by defining the NameNode and DataNode storage directories. In this "hdfs-site.xml" file, we will change the directory path of "datanode" and "namenode": Additionally, the default dfs.replication value of 3 needs to be changed to 1 to match the single node setup.

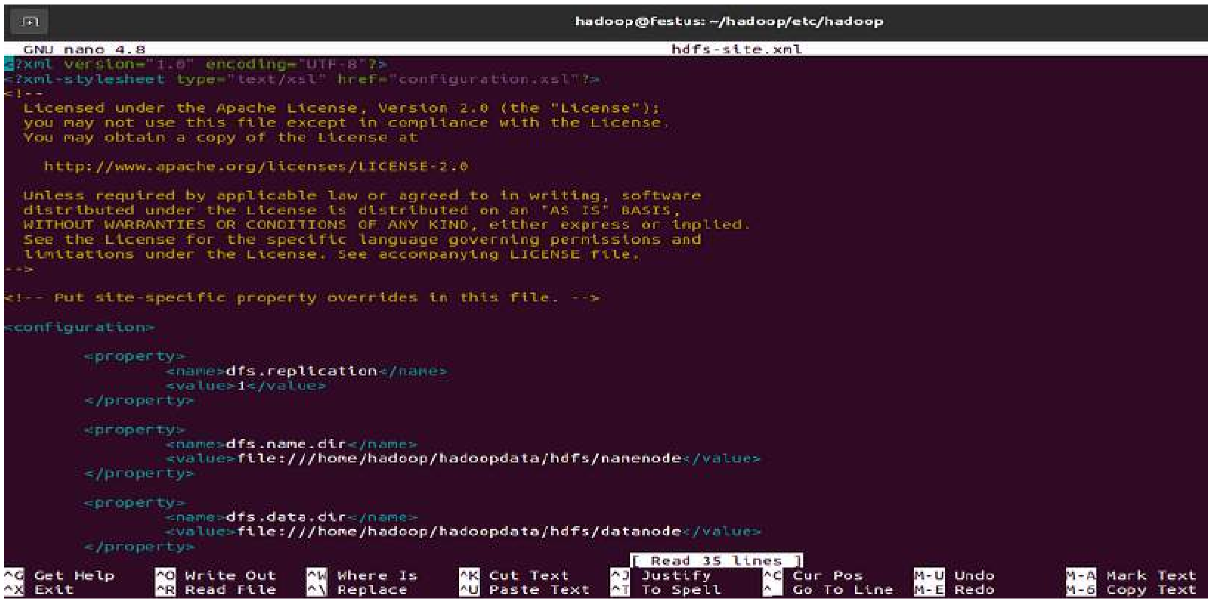
Use the following command to open the hdfs-site.xml file for editing:

sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml

Add the following configuration to the file and, if needed, adjust the NameNode and DataNode directories to your custom locations:

```
<configuration>
<property>
          <name>dfs.replication</name>
          <value>1</value>
     </property>
<property>
          <name>dfs.name.dir</name>
          <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
     </property>
<property>
          <name>dfs.data.dir</name>
          <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
     </property>
</configuration>
```

If necessary, create the specific directories you defined for the dfs.data.dir value.



```
hadoop@festus: ~/hadoop/etc/hadoop
  GNU nano 4.8                                      hdfs-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

        <property>
                <name>dfs.replication</name>
                <value>1</value>
        </property>

        <property>
                <name>dfs.name.dir</name>
                <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
        </property>

        <property>
                <name>dfs.data.dir</name>
                <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
        </property>
                                          [ Read 35 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo      M-A Mark Text
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line M-E Redo      M-6 Copy Text
```

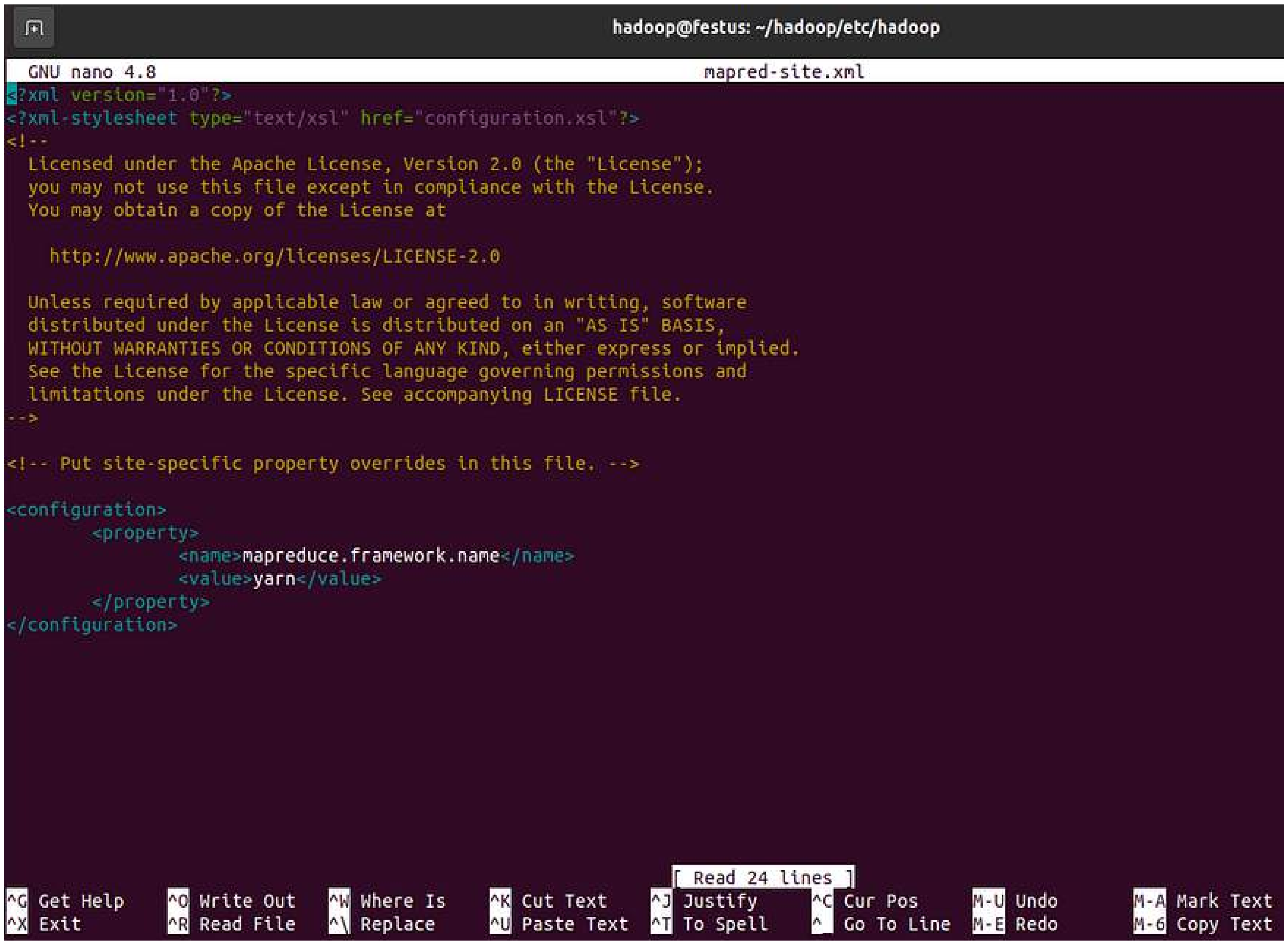
**Step 5e: Edit mapred-site.xml File**

Use the following command to access the mapred-site.xml file and define MapReduce values:

sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml

Add the following configuration to change the default MapReduce framework name value to yarn:

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>
```



```
hadoop@festus: ~/hadoop/etc/hadoop
  GNU nano 4.8                                   mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>mapreduce.framework.name</name>
                <value>yarn</value>
        </property>
</configuration>

                                          [ Read 24 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo      M-A Mark Text
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line M-E Redo      M-6 Copy Text
```

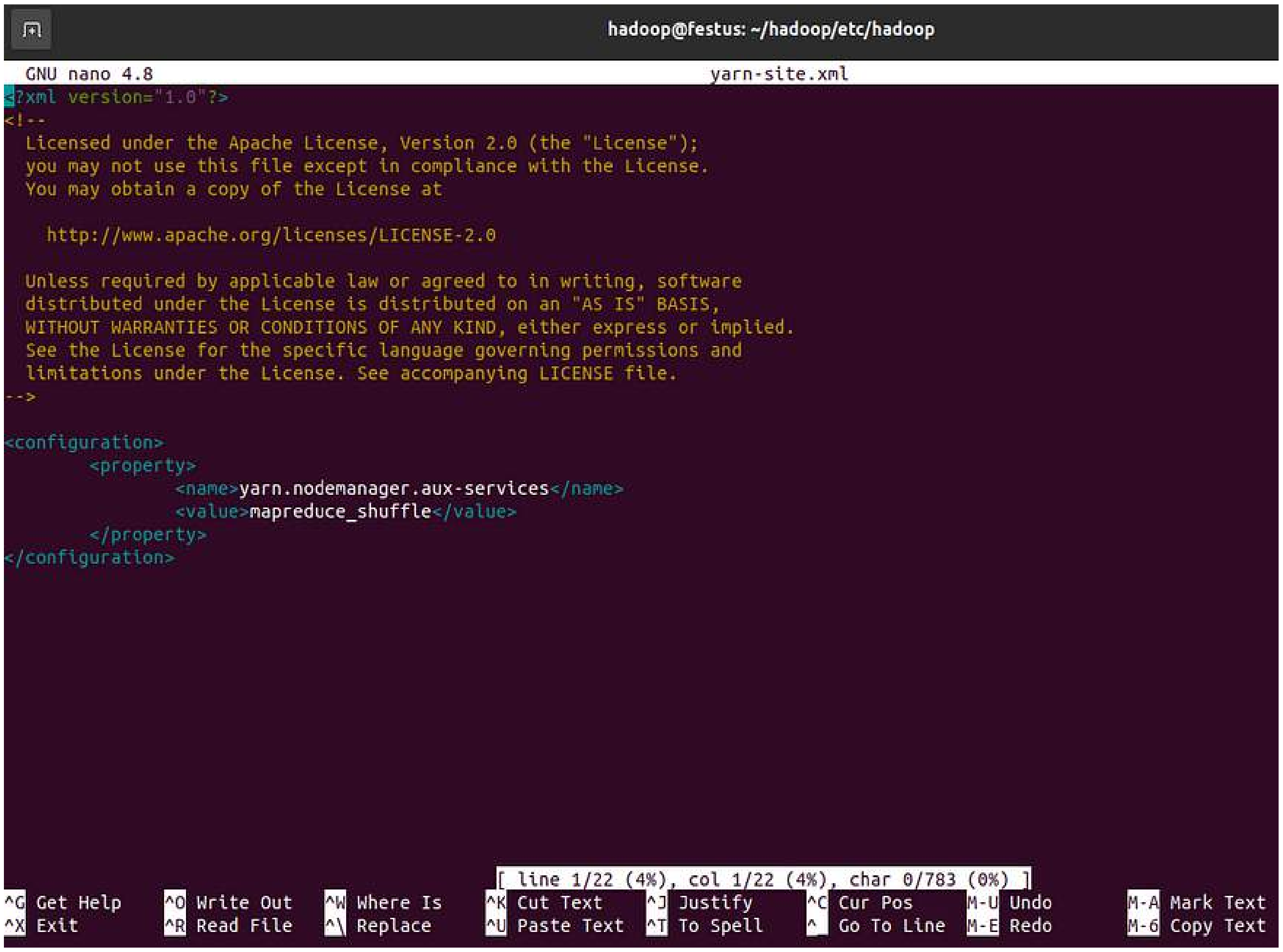
**Step 5f: Edit yarn-site.xml File**

The yarn-site.xml file is used to define settings relevant to YARN. It contains configurations for the Node Manager, Resource Manager, Containers, and Application Master. Open the yarn-site.xml file in a text editor:

sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml

Append the following configuration to the file:

```
<configuration>
        <property>
                <name>yarn.nodemanager.aux-services</name>
                <value>mapreduce_shuffle</value>
        </property>
</configuration>
```

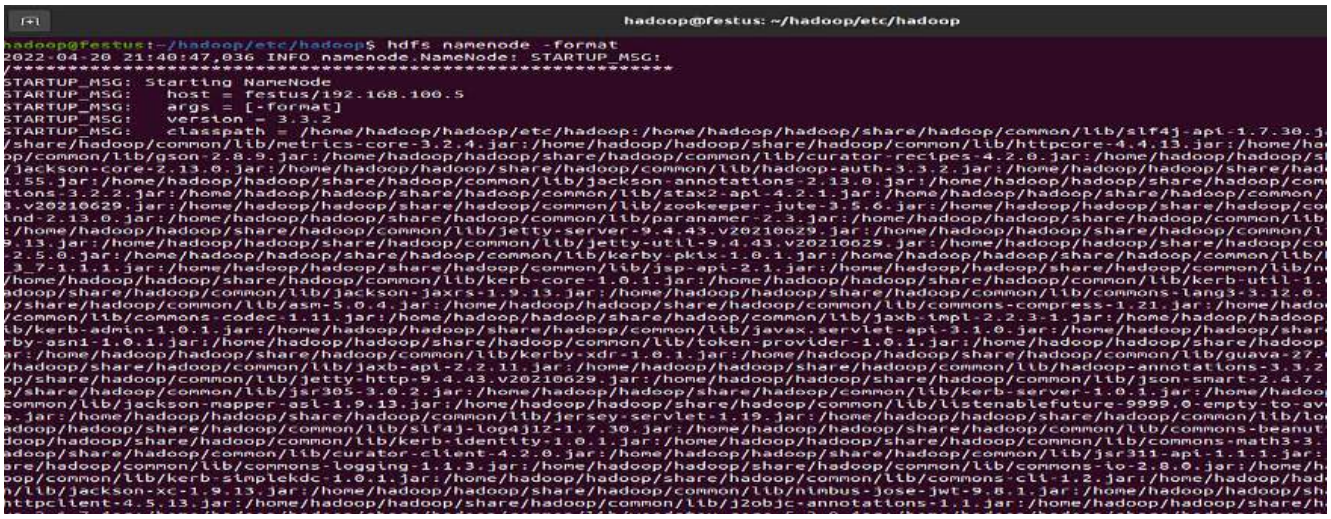


```
hadoop@festus: ~/hadoop/etc/hadoop
  GNU nano 4.8                                              yarn-site.xml
<?xml version="1.0"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<configuration>
        <property>
                <name>yarn.nodemanager.aux-services</name>
                <value>mapreduce_shuffle</value>
        </property>
</configuration>

                                              [ line 1/22 (4%), col 1/22 (4%), char 0/783 (0%) ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo      M-A Mark Text
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line M-E Redo      M-6 Copy Text
```

**Step 5g. Format HDFS NameNode**

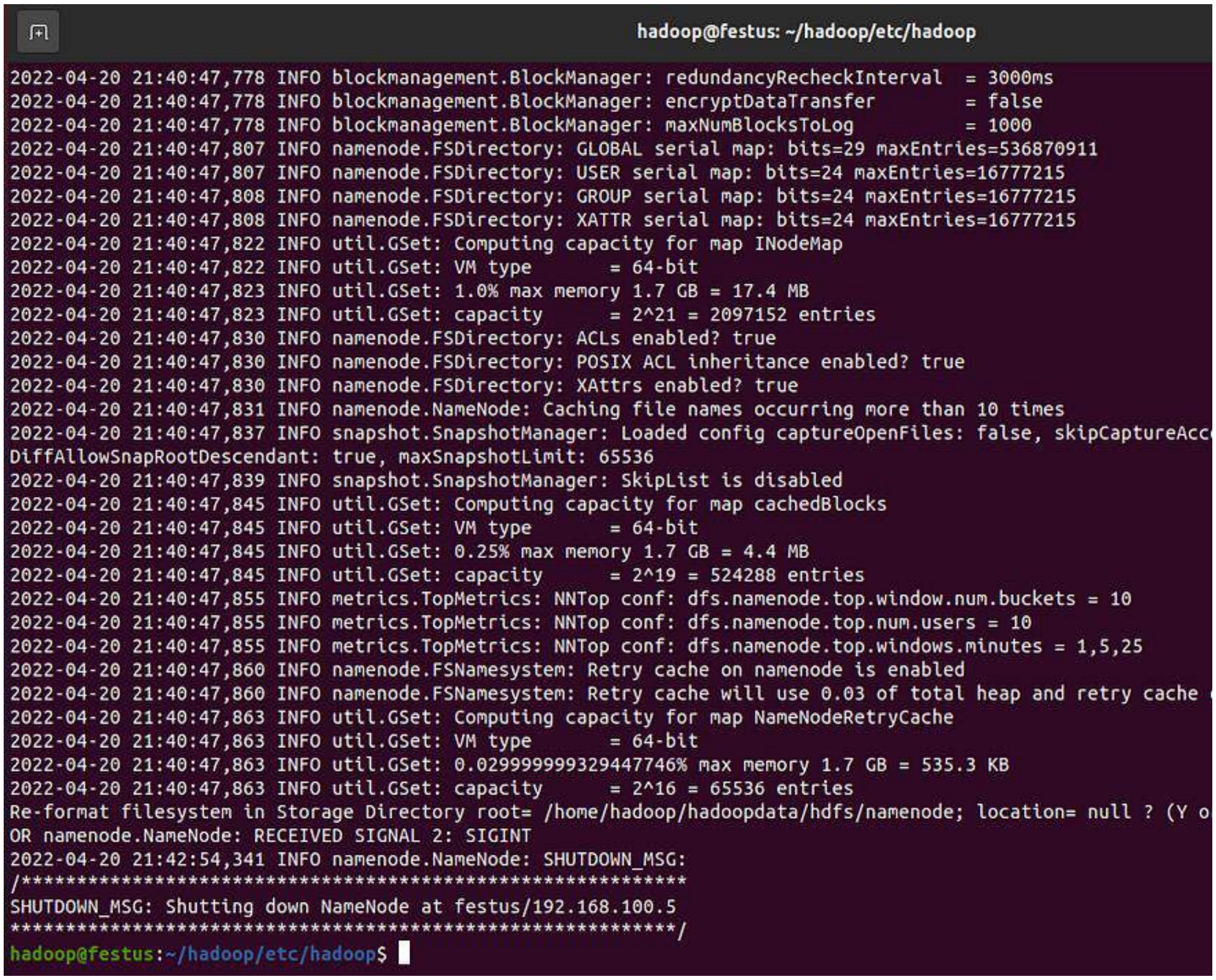
It is important to format the NameNode before starting Hadoop services for the first time:

hdfs namenode -format



```
hadoop@festus:~/hadoop/etc/hadoop$ hdfs namenode -format
2022-04-20 21:40:47,036 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = festus/192.168.100.5
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.2
STARTUP_MSG:   classpath = /home/hadoop/hadoop/etc/hadoop:/home/hadoop/hadoop/share/hadoop/common/lib/slf4j-api-1.7.30.j
/share/hadoop/common/lib/metrics-core-3.2.4.jar:/home/hadoop/hadoop/share/hadoop/common/lib/httpcore-4.4.13.jar:/home/ha
op/common/lib/gson-2.8.9.jar:/home/hadoop/hadoop/share/hadoop/common/lib/curator-recipes-4.2.0.jar:/home/hadoop/hadoop/s
/jackson-core-2.13.0.jar:/home/hadoop/hadoop/share/hadoop/common/lib/hadoop-auth-3.3.2.jar:/home/hadoop/hadoop/share/had
1.55.jar:/home/hadoop/hadoop/share/hadoop/common/lib/jackson-annotations-2.13.0.jar:/home/hadoop/hadoop/share/hadoop/com
tions-3.2.2.jar:/home/hadoop/hadoop/share/hadoop/common/lib/stax2-api-4.2.1.jar:/home/hadoop/hadoop/share/hadoop/common/
3.v20210629.jar:/home/hadoop/hadoop/share/hadoop/common/lib/zookeeper-jute-3.5.6.jar:/home/hadoop/hadoop/share/hadoop/co
ind-2.13.0.jar:/home/hadoop/hadoop/share/hadoop/common/lib/paranamer-2.3.jar:/home/hadoop/hadoop/share/hadoop/common/lib
:/home/hadoop/hadoop/share/hadoop/common/lib/jetty-server-9.4.43.v20210629.jar:/home/hadoop/hadoop/share/hadoop/common/l
9.13.jar:/home/hadoop/hadoop/share/hadoop/common/lib/jetty-util-9.4.43.v20210629.jar:/home/hadoop/hadoop/share/hadoop/co
-2.5.0.jar:/home/hadoop/hadoop/share/hadoop/common/lib/kerby-pkix-1.0.1.jar:/home/hadoop/hadoop/share/hadoop/common/lib/
_3_7-1.1.1.jar:/home/hadoop/hadoop/share/hadoop/common/lib/jsp-api-2.1.jar:/home/hadoop/hadoop/share/hadoop/common/lib/n
/home/hadoop/hadoop/share/hadoop/common/lib/kerb-core-1.0.1.jar:/home/hadoop/hadoop/share/hadoop/common/lib/kerb-util-1.
adoop/share/hadoop/common/lib/jackson-jaxrs-1.9.13.jar:/home/hadoop/hadoop/share/hadoop/common/lib/commons-lang3-3.12.0.
p/share/hadoop/common/lib/asm-5.0.4.jar:/home/hadoop/hadoop/share/hadoop/common/lib/commons-compress-1.21.jar:/home/hado
/common/lib/commons-codec-1.11.jar:/home/hadoop/hadoop/share/hadoop/common/lib/jaxb-impl-2.2.3-1.jar:/home/hadoop/hadoop
ib/kerb-admin-1.0.1.jar:/home/hadoop/hadoop/share/hadoop/common/lib/javax.servlet-api-3.1.0.jar:/home/hadoop/hadoop/shar
rby-asn1-1.0.1.jar:/home/hadoop/hadoop/share/hadoop/common/lib/token-provider-1.0.1.jar:/home/hadoop/hadoop/share/hadoop
ar:/home/hadoop/hadoop/share/hadoop/common/lib/kerby-xdr-1.0.1.jar:/home/hadoop/hadoop/share/hadoop/common/lib/guava-27.
/hadoop/share/hadoop/common/lib/jaxb-api-2.2.11.jar:/home/hadoop/hadoop/share/hadoop/common/lib/hadoop-annotations-3.3.2
op/share/hadoop/common/lib/jetty-http-9.4.43.v20210629.jar:/home/hadoop/hadoop/share/hadoop/common/lib/json-smart-2.4.7.
p/share/hadoop/common/lib/jsr305-3.0.2.jar:/home/hadoop/hadoop/share/hadoop/common/lib/kerb-server-1.0.1.jar:/home/hadoo
common/lib/jackson-mapper-asl-1.9.13.jar:/home/hadoop/hadoop/share/hadoop/common/lib/listenablefuture-9999.0-empty-to-av
a.jar:/home/hadoop/hadoop/share/hadoop/common/lib/jersey-servlet-1.19.jar:/home/hadoop/hadoop/share/hadoop/common/lib/lo
adoop/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar:/home/hadoop/hadoop/share/hadoop/common/lib/commons-beanut
doop/hadoop/share/hadoop/common/lib/kerb-identity-1.0.1.jar:/home/hadoop/hadoop/share/hadoop/common/lib/commons-math3-3.
adoop/share/hadoop/common/lib/curator-client-4.2.0.jar:/home/hadoop/hadoop/share/hadoop/common/lib/jsr311-api-1.1.1.jar:
are/hadoop/common/lib/commons-logging-1.1.3.jar:/home/hadoop/hadoop/share/hadoop/common/lib/commons-io-2.8.0.jar:/home/h
oop/common/lib/kerb-simplekdc-1.0.1.jar:/home/hadoop/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar:/home/hadoop/had
n/lib/jackson-xc-1.9.13.jar:/home/hadoop/hadoop/share/hadoop/common/lib/nimbus-jose-jwt-9.8.1.jar:/home/hadoop/hadoop/sh
httpclient-4.5.13.jar:/home/hadoop/hadoop/share/hadoop/common/lib/j2objc-annotations-1.1.jar:/home/hadoop/hadoop/share/h
```

The shutdown notification signifies the end of the NameNode format process.

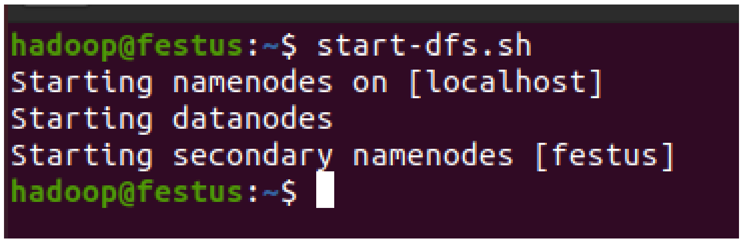


```
2022-04-20 21:40:47,778 INFO blockmanagement.BlockManager: redundancyRecheckInterval  = 3000ms
2022-04-20 21:40:47,778 INFO blockmanagement.BlockManager: encryptDataTransfer        = false
2022-04-20 21:40:47,778 INFO blockmanagement.BlockManager: maxNumBlocksToLog          = 1000
2022-04-20 21:40:47,807 INFO namenode.FSDirectory: GLOBAL serial map: bits=29 maxEntries=536870911
2022-04-20 21:40:47,807 INFO namenode.FSDirectory: USER serial map: bits=24 maxEntries=16777215
2022-04-20 21:40:47,808 INFO namenode.FSDirectory: GROUP serial map: bits=24 maxEntries=16777215
2022-04-20 21:40:47,808 INFO namenode.FSDirectory: XATTR serial map: bits=24 maxEntries=16777215
2022-04-20 21:40:47,822 INFO util.GSet: Computing capacity for map INodeMap
2022-04-20 21:40:47,822 INFO util.GSet: VM type       = 64-bit
2022-04-20 21:40:47,823 INFO util.GSet: 1.0% max memory 1.7 GB = 17.4 MB
2022-04-20 21:40:47,823 INFO util.GSet: capacity      = 2^21 = 2097152 entries
2022-04-20 21:40:47,830 INFO namenode.FSDirectory: ACLs enabled? true
2022-04-20 21:40:47,830 INFO namenode.FSDirectory: POSIX ACL inheritance enabled? true
2022-04-20 21:40:47,830 INFO namenode.FSDirectory: XAttrs enabled? true
2022-04-20 21:40:47,831 INFO namenode.NameNode: Caching file names occurring more than 10 times
2022-04-20 21:40:47,837 INFO snapshot.SnapshotManager: Loaded config captureOpenFiles: false, skipCaptureAcc
DiffAllowSnapRootDescendant: true, maxSnapshotLimit: 65536
2022-04-20 21:40:47,839 INFO snapshot.SnapshotManager: SkipList is disabled
2022-04-20 21:40:47,845 INFO util.GSet: Computing capacity for map cachedBlocks
2022-04-20 21:40:47,845 INFO util.GSet: VM type       = 64-bit
2022-04-20 21:40:47,845 INFO util.GSet: 0.25% max memory 1.7 GB = 4.4 MB
2022-04-20 21:40:47,845 INFO util.GSet: capacity      = 2^19 = 524288 entries
2022-04-20 21:40:47,855 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2022-04-20 21:40:47,855 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2022-04-20 21:40:47,855 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2022-04-20 21:40:47,860 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2022-04-20 21:40:47,860 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache
2022-04-20 21:40:47,863 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2022-04-20 21:40:47,863 INFO util.GSet: VM type       = 64-bit
2022-04-20 21:40:47,863 INFO util.GSet: 0.029999999329447746% max memory 1.7 GB = 535.3 KB
2022-04-20 21:40:47,863 INFO util.GSet: capacity      = 2^16 = 65536 entries
Re-format filesystem in Storage Directory root= /home/hadoop/hadoopdata/hdfs/namenode; location= null ? (Y o
OR namenode.NameNode: RECEIVED SIGNAL 2: SIGINT
2022-04-20 21:42:54,341 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at festus/192.168.100.5
************************************************************/
hadoop@festus:~/hadoop/etc/hadoop$ 
```

Step 6: Start Hadoop Cluster

Start the NameNode and DataNode.

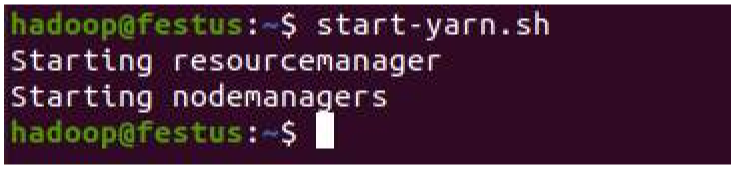
$ start-dfs.sh



```
hadoop@festus:~$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [festus]
hadoop@festus:~$
```

Start the YARN resource and node managers.

$ start-yarn.sh

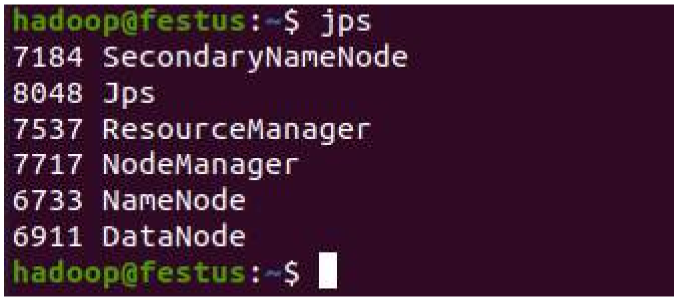


```
hadoop@festus:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@festus:~$
```

Verify all the running components.

$ jps

The system takes a few moments to initiate the necessary nodes. If everything is working as intended, the resulting list of running Java processes contains all the HDFS and YARN daemons.



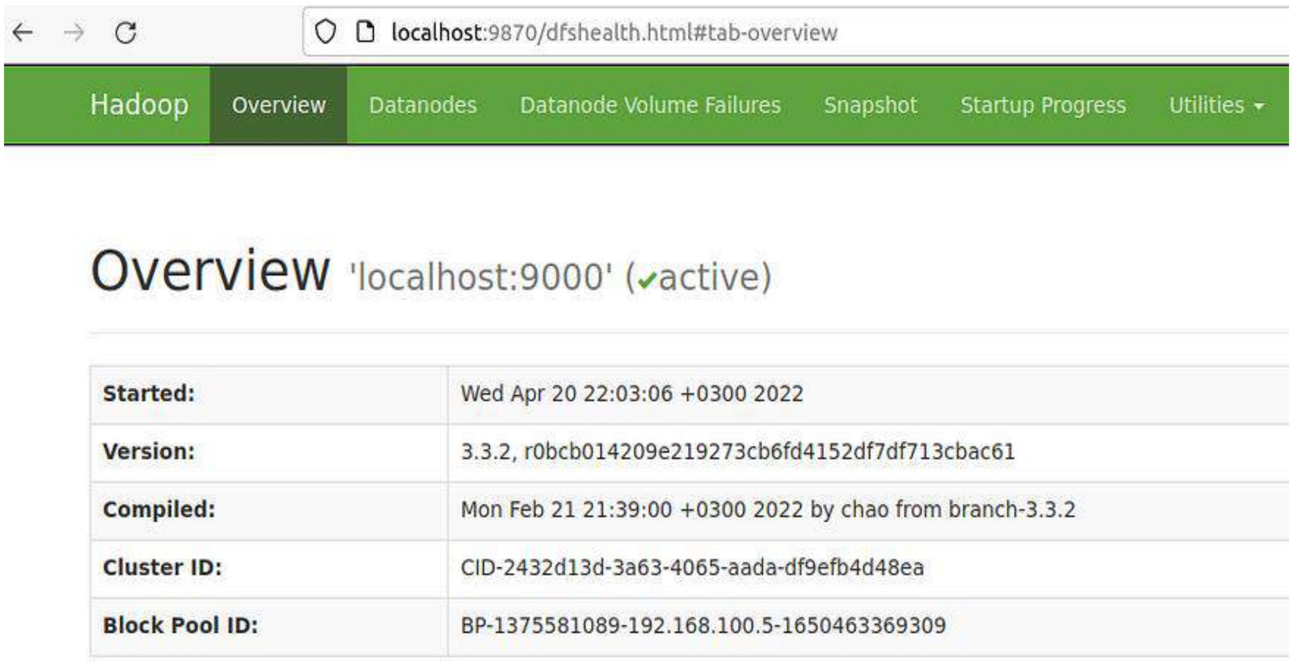
```
hadoop@festus:~$ jps
7184 SecondaryNameNode
8048 Jps
7537 ResourceManager
7717 NodeManager
6733 NameNode
6911 DataNode
hadoop@festus:~$
```

Step 7: Access Hadoop UI from Browser

Use your preferred browser and navigate to your localhost URL or IP. The default port number 9870 gives you access to the Hadoop NameNode UI:

http://localhost:9870

The NameNode user interface provides a comprehensive overview of the entire cluster

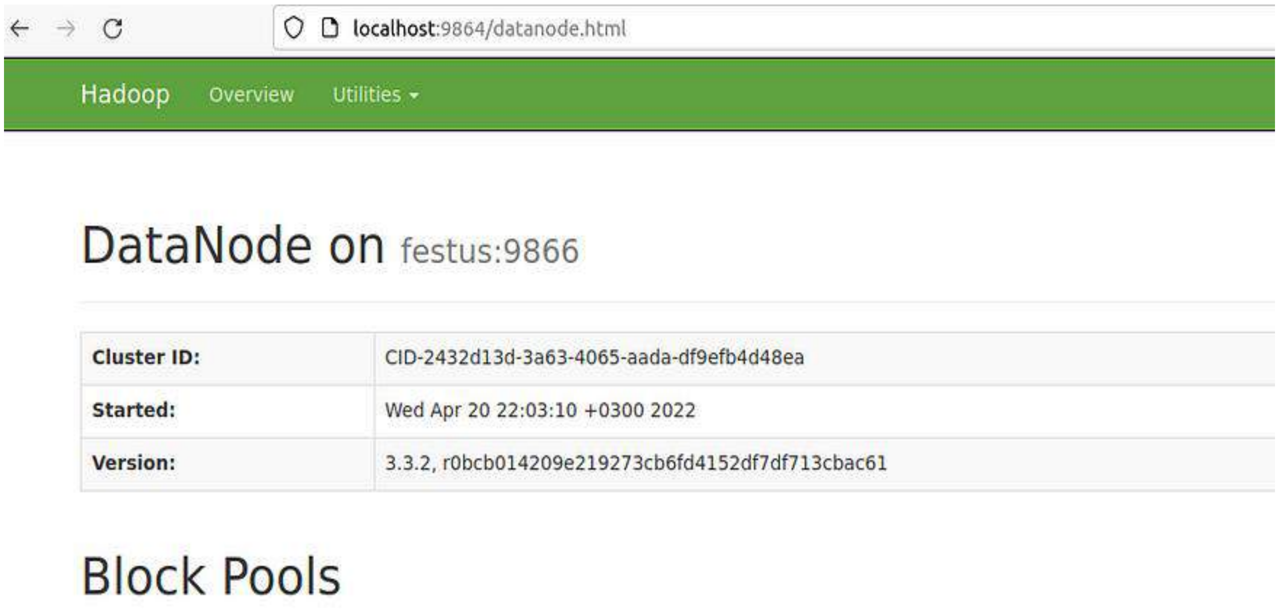


Overview 'localhost:9000' (✓active)

| | |
|---|---|
| Started: | Wed Apr 20 22:03:06 +0300 2022 |
| Version: | 3.3.2, r0bcb014209e219273cb6fd4152df7df713cbac61 |
| Compiled: | Mon Feb 21 21:39:00 +0300 2022 by chao from branch-3.3.2 |
| Cluster ID: | CID-2432d13d-3a63-4065-aada-df9efb4d48ea |
| Block Pool ID: | BP-1375581089-192.168.100.5-1650463369309 |

The default port 9864 is used to access individual DataNodes directly from your browser:

http://localhost:9864



DataNode on festus:9866

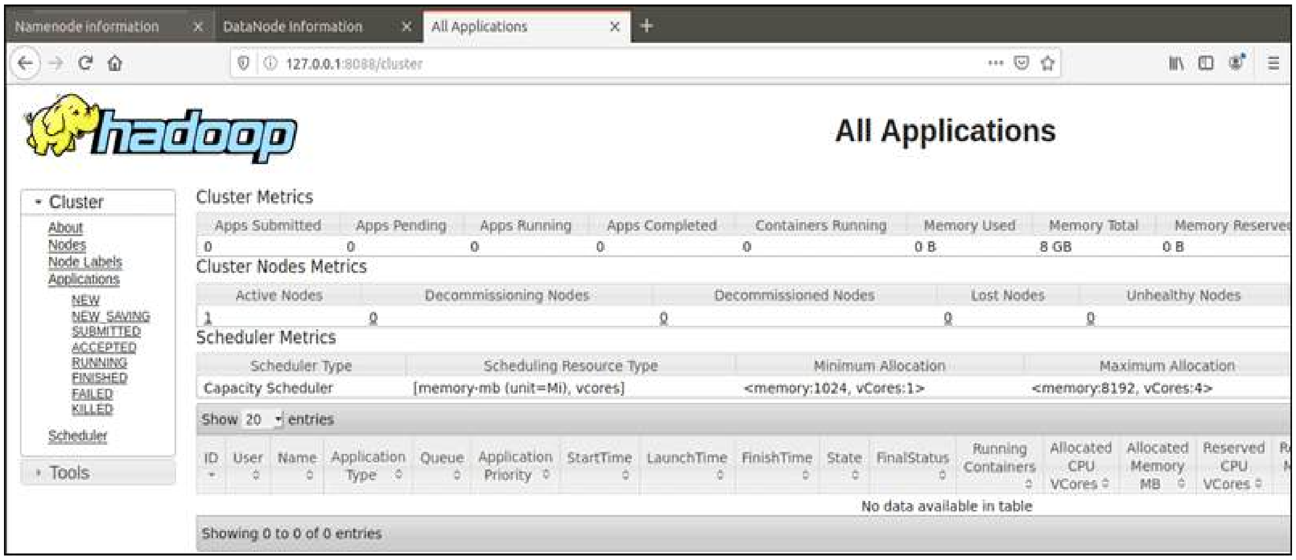
| | |
|---|---|
| Cluster ID: | CID-2432d13d-3a63-4065-aada-df9efb4d48ea |
| Started: | Wed Apr 20 22:03:10 +0300 2022 |
| Version: | 3.3.2, r0bcb014209e219273cb6fd4152df7df713cbac61 |

Block Pools

The YARN Resource Manager is accessible on port 8088:

http://localhost:8088

The Resource Manager is an invaluable tool that allows you to monitor all running processes in your Hadoop cluster.



All Applications

Cluster Metrics

| Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Memory Used | Memory Total | Memory Reserve |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 B | 8 GB | 0 B |

Cluster Nodes Metrics

| Active Nodes | Decommissioning Nodes | Decommissioned Nodes | Lost Nodes | Unhealthy Nodes |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |

Scheduler Metrics

| Scheduler Type | Scheduling Resource Type | Minimum Allocation | Maximum Allocation |
|---|---|---|---|
| Capacity Scheduler | [memory-mb (unit=Mi), vcores] | <memory:1024, vCores:1> | <memory:8192, vCores:4> |

Show 20 entries

| ID | User | Name | Application Type | Queue | Application Priority | StartTime | LaunchTime | FinishTime | State | FinalStatus | Running Containers | Allocated CPU VCores | Allocated Memory MB | Reserved CPU VCores | R M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No data available in table | | | | | | | | | | | | | | | |

Showing 0 to 0 of 0 entries

## Result

Thus, the step by step Installing Hadoop on Ubuntu 20.04 was successfully done.

## Ex.No.2 Implement Word count / frequency program using Map reduce

**Aim**

To implement Word count / frequency program using Map reduce.

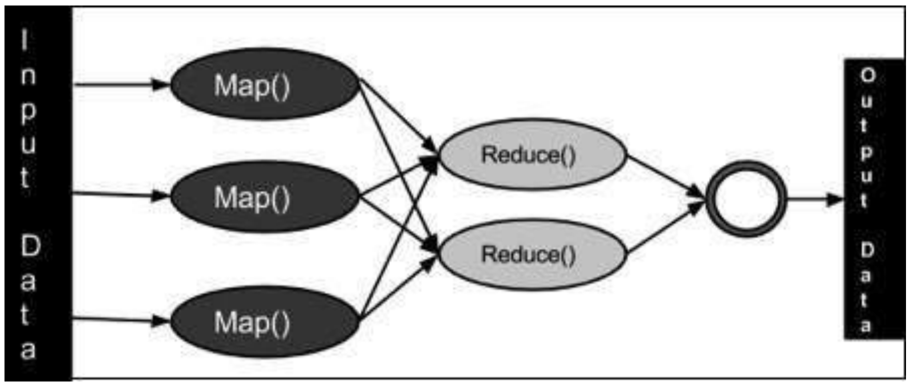
**Pre-Lab discussion**

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into *mappers* and *reducers* is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

The Algorithm

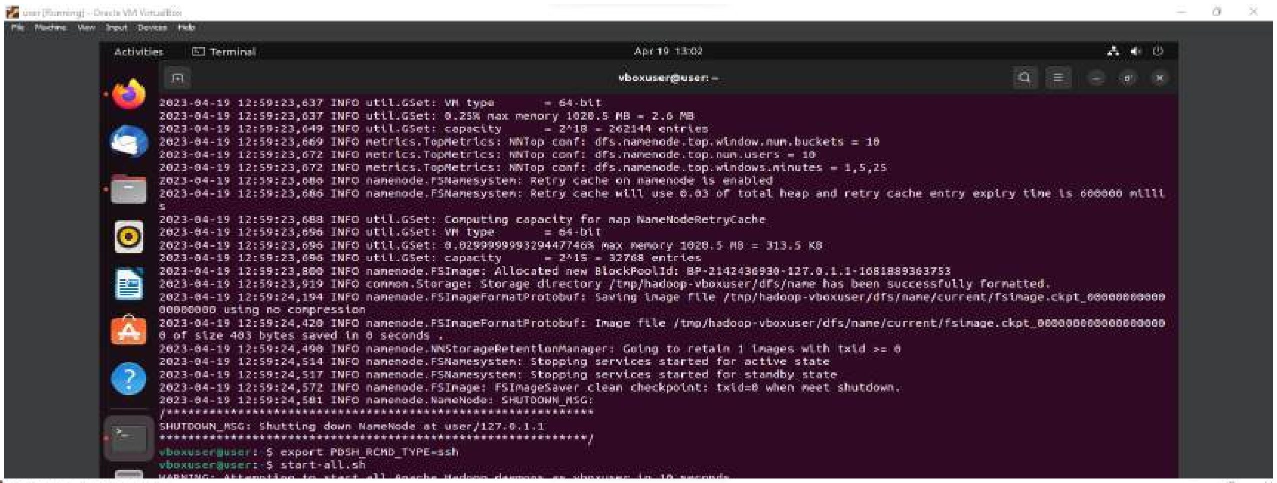
- Generally MapReduce paradigm is based on sending the computer to where the data resides!
- MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.
    - **Map stage** − The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
    - **Reduce stage** − This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.
- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.
- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.



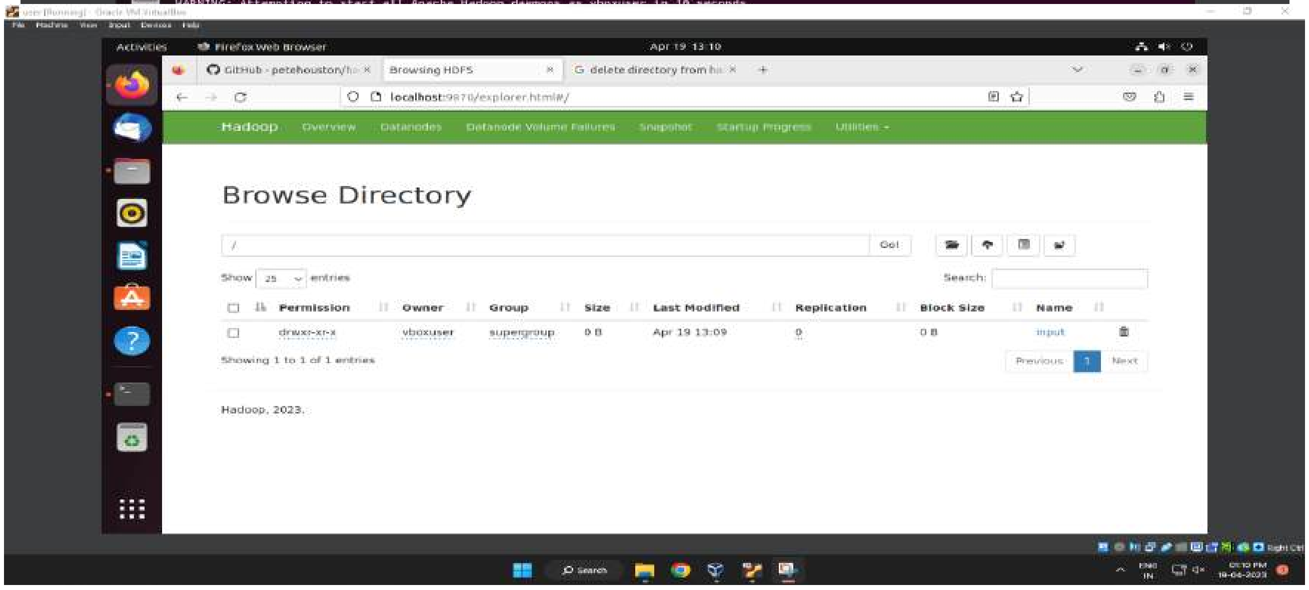
## Steps on Hadoop Operations

Step 1: Start

$ start-all.sh



```
2023-04-19 12:59:23,637 INFO util.GSet: VM type       = 64-bit
2023-04-19 12:59:23,637 INFO util.GSet: 0.25% max memory 1020.5 MB = 2.6 MB
2023-04-19 12:59:23,649 INFO util.GSet: capacity      = 2^18 = 262144 entries
2023-04-19 12:59:23,669 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2023-04-19 12:59:23,672 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2023-04-19 12:59:23,672 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2023-04-19 12:59:23,686 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2023-04-19 12:59:23,686 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 milli
s
2023-04-19 12:59:23,688 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2023-04-19 12:59:23,696 INFO util.GSet: VM type       = 64-bit
2023-04-19 12:59:23,696 INFO util.GSet: 0.029999999329447746% max memory 1020.5 MB = 313.5 KB
2023-04-19 12:59:23,696 INFO util.GSet: capacity      = 2^15 = 32768 entries
2023-04-19 12:59:23,800 INFO namenode.FSImage: Allocated new BlockPoolId: BP-2142436930-127.0.1.1-1681889363753
2023-04-19 12:59:23,919 INFO common.Storage: Storage directory /tmp/hadoop-vboxuser/dfs/name has been successfully formatted.
2023-04-19 12:59:24,194 INFO namenode.FSImageFormatProtobuf: Saving image file /tmp/hadoop-vboxuser/dfs/name/current/fsimage.ckpt_00000000000
00000000 using no compression
2023-04-19 12:59:24,420 INFO namenode.FSImageFormatProtobuf: Image file /tmp/hadoop-vboxuser/dfs/name/current/fsimage.ckpt_000000000000000000
0 of size 403 bytes saved in 0 seconds .
2023-04-19 12:59:24,490 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2023-04-19 12:59:24,514 INFO namenode.FSNamesystem: Stopping services started for active state
2023-04-19 12:59:24,517 INFO namenode.FSNamesystem: Stopping services started for standby state
2023-04-19 12:59:24,572 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2023-04-19 12:59:24,581 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at user/127.0.1.1
************************************************************/
vboxuser@user:~$ export PDSH_RCMD_TYPE=ssh
vboxuser@user:~$ start-all.sh
```



localhost:9870/explorer.html#/

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

### Browse Directory

/ Go!

Show 25 entries Search:

| Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|
| drwxr-xr-x | vboxuser | supergroup | 0 B | Apr 19 13:09 | 0 | 0 B | input |

Showing 1 to 1 of 1 entries

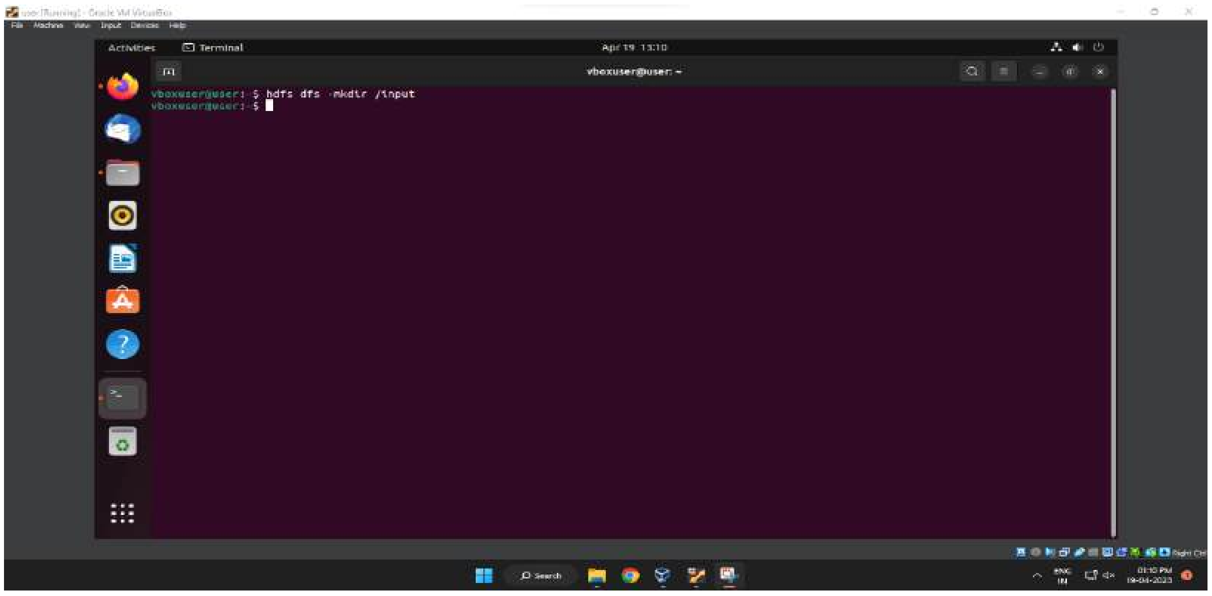
Previous 1 Next

Hadoop, 2023.

Step 2 : Start output

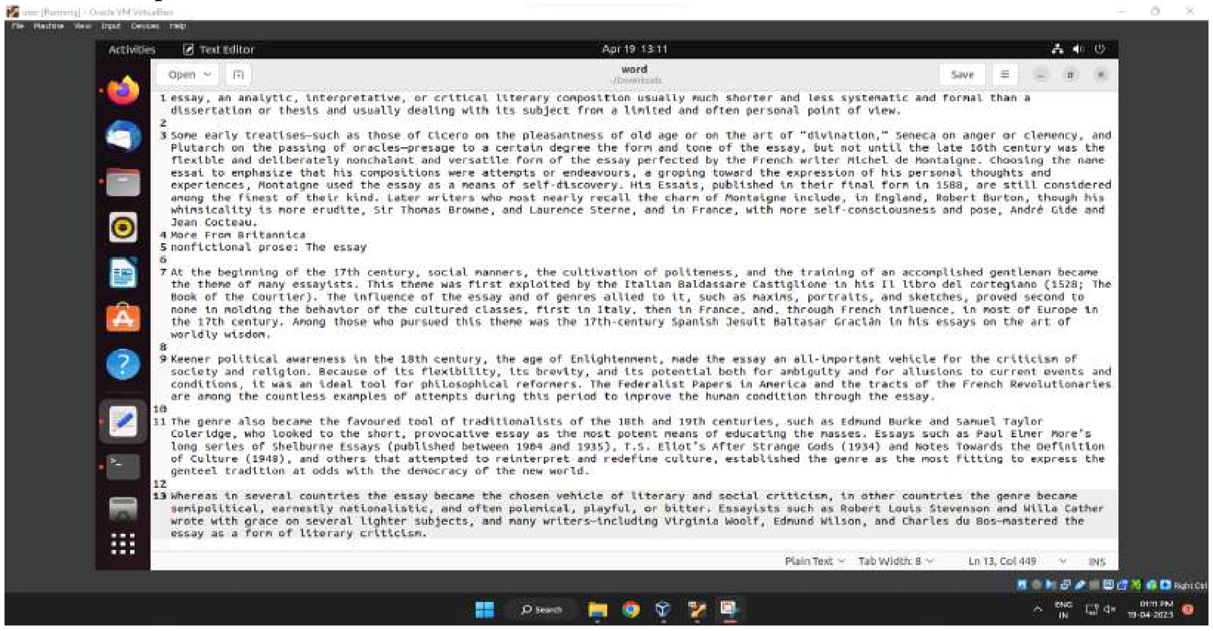
Step 3 :Create an input directory

$ hdfs dfs –mkdir /input



```
vboxuser@user:~$ hdfs dfs -mkdir /input
vboxuser@user:~$ 
```

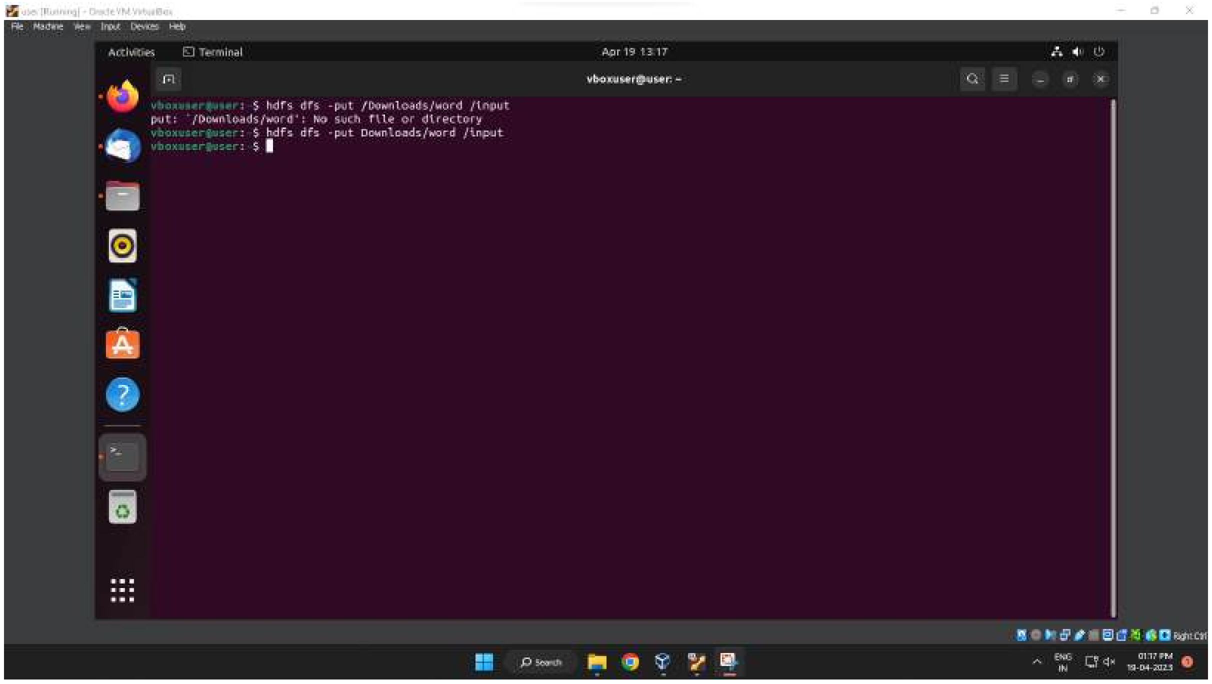
Step 4 : Create a sample word



```
essay, an analytic, interpretative, or critical literary composition usually much shorter and less systematic and formal than a
dissertation or thesis and usually dealing with its subject from a limited and often personal point of view.

Some early treatises—such as those of Cicero on the pleasantness of old age or on the art of "divination," Seneca on anger or clemency, and
Plutarch on the passing of oracles—presage to a certain degree the form and tone of the essay, but not until the late 16th century was the
flexible and deliberately nonchalant and versatile form of the essay perfected by the French writer Michel de Montaigne. Choosing the name
essai to emphasize that his compositions were attempts or endeavours, a groping toward the expression of his personal thoughts and
experiences, Montaigne used the essay as a means of self-discovery. His Essais, published in their final form in 1588, are still considered
among the finest of their kind. Later writers who most nearly recall the charm of Montaigne include, in England, Robert Burton, though his
whimsicality is more erudite, Sir Thomas Browne, and Laurence Sterne, and in France, with more self-consciousness and pose, André Gide and
Jean Cocteau.
More From Britannica
nonfictional prose: The essay

At the beginning of the 17th century, social manners, the cultivation of politeness, and the training of an accomplished gentleman became
the theme of many essayists. This theme was first exploited by the Italian Baldassare Castiglione in his Il libro del cortegiano (1528; The
Book of the Courtier). The influence of the essay and of genres allied to it, such as maxims, portraits, and sketches, proved second to
none in molding the behavior of the cultured classes, first in Italy, then in France, and, through French influence, in most of Europe in
the 17th century. Among those who pursued this theme was the 17th-century Spanish Jesuit Baltasar Gracián in his essays on the art of
worldly wisdom.

Keener political awareness in the 18th century, the age of Enlightenment, made the essay an all-important vehicle for the criticism of
society and religion. Because of its flexibility, its brevity, and its potential both for ambiguity and for allusions to current events and
conditions, it was an ideal tool for philosophical reformers. The Federalist Papers in America and the tracts of the French Revolutionaries
are among the countless examples of attempts during this period to improve the human condition through the essay.

The genre also became the favoured tool of traditionalists of the 18th and 19th centuries, such as Edmund Burke and Samuel Taylor
Coleridge, who looked to the short, provocative essay as the most potent means of educating the masses. Essays such as Paul Elmer More's
long series of Shelburne Essays (published between 1904 and 1935), T.S. Eliot's After Strange Gods (1934) and Notes Towards the Definition
of Culture (1948), and others that attempted to reinterpret and redefine culture, established the genre as the most fitting to express the
genteel tradition at odds with the democracy of the new world.

Whereas in several countries the essay became the chosen vehicle of literary and social criticism, in other countries the genre became
semipolitical, earnestly nationalistic, and often polemical, playful, or bitter. Essayists such as Robert Louis Stevenson and Willa Cather
wrote with grace on several lighter subjects, and many writers—including Virginia Woolf, Edmund Wilson, and Charles du Bos—mastered the
essay as a form of literary criticism.
```

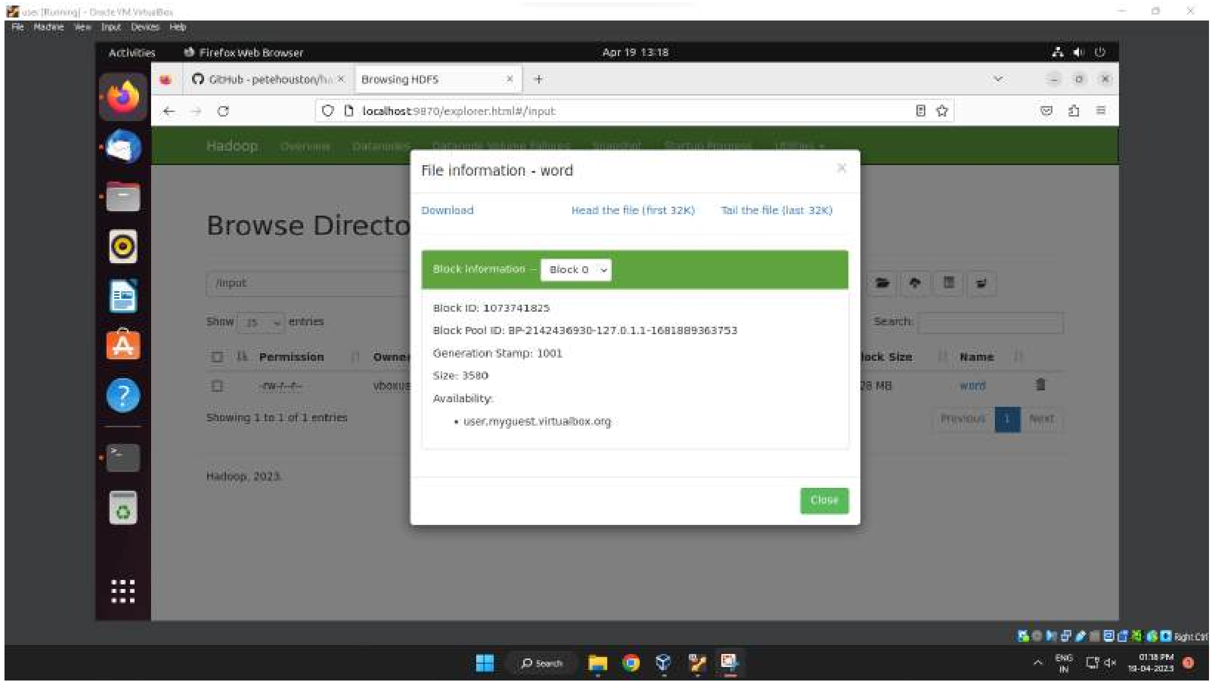
Step 5: Put file

$ hdfs dfs –put Downloads/word /input



```
vboxuser@user:~$ hdfs dfs -put /Downloads/word /input
put: `/Downloads/word': No such file or directory
vboxuser@user:~$ hdfs dfs -put Downloads/word /input
vboxuser@user:~$ 
```

Step 6 : Put file output



localhost:9870/explorer.html#/input

File information - word

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

Block information – Block 0

Block ID: 1073741825

Block Pool ID: BP-2142436930-127.0.1.1-1681889363753

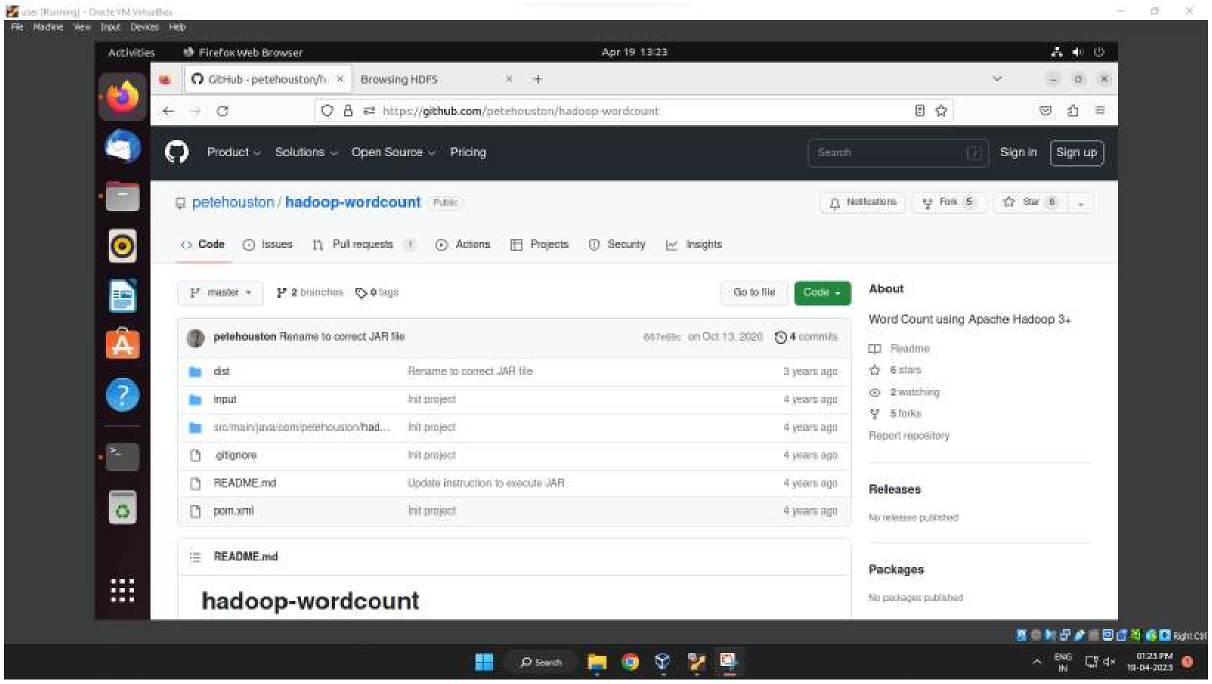
Generation Stamp: 1001

Size: 3580

Availability:

- user.myguest.virtualbox.org

Step 7 : Downloading jar



petehouston / hadoop-wordcount Public

| | | |
|---|---|---|
| dist | Rename to correct JAR file | 3 years ago |
| input | Init project | 4 years ago |
| src/main/java/com/petehouston/had... | Init project | 4 years ago |
| .gitignore | Init project | 4 years ago |
| README.md | Update instruction to execute JAR | 4 years ago |
| pom.xml | Init project | 4 years ago |

About

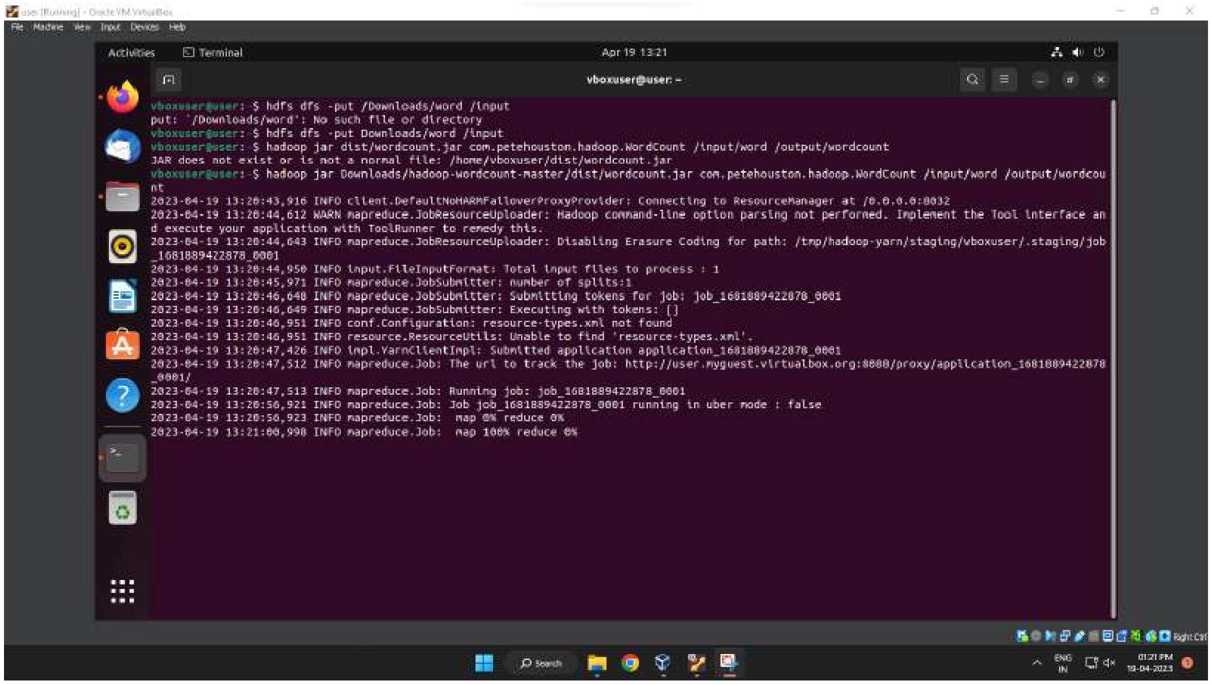
Word Count using Apache Hadoop 3+

hadoop-wordcount

Step 8: Loading jar

$ Hadoop jar Downloads/Hadoop-wordcount-master/dist/wordcount.jar

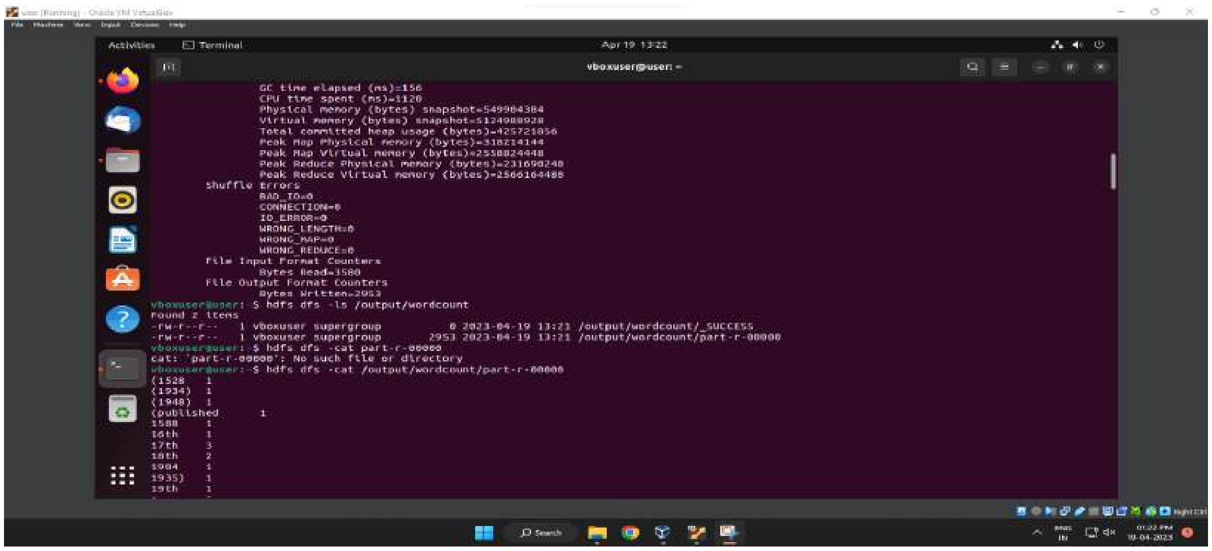
com.petehouston.hadoop.WordCount /input/word/output



```
vboxuser@user:~$ hdfs dfs -put /Downloads/word /input
put: `/Downloads/word': No such file or directory
vboxuser@user:~$ hdfs dfs -put Downloads/word /input
vboxuser@user:~$ hadoop jar dist/wordcount.jar com.petehouston.hadoop.WordCount /input/word /output/wordcount
JAR does not exist or is not a normal file: /home/vboxuser/dist/wordcount.jar
vboxuser@user:~$ hadoop jar Downloads/hadoop-wordcount-master/dist/wordcount.jar com.petehouston.hadoop.WordCount /input/word /output/wordcou
nt
2023-04-19 13:20:43,916 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2023-04-19 13:20:44,612 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface an
d execute your application with ToolRunner to remedy this.
2023-04-19 13:20:44,643 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/vboxuser/.staging/job
_1681889422878_0001
2023-04-19 13:20:44,950 INFO input.FileInputFormat: Total input files to process : 1
2023-04-19 13:20:45,971 INFO mapreduce.JobSubmitter: number of splits:1
2023-04-19 13:20:46,640 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1681889422878_0001
2023-04-19 13:20:46,649 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-04-19 13:20:46,951 INFO conf.Configuration: resource-types.xml not found
2023-04-19 13:20:46,951 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-04-19 13:20:47,426 INFO impl.YarnClientImpl: Submitted application application_1681889422878_0001
2023-04-19 13:20:47,512 INFO mapreduce.Job: The url to track the job: http://user.myguest.virtualbox.org:8088/proxy/application_1681889422878
_0001/
2023-04-19 13:20:47,513 INFO mapreduce.Job: Running job: job_1681889422878_0001
2023-04-19 13:20:56,921 INFO mapreduce.Job: Job job_1681889422878_0001 running in uber mode : false
2023-04-19 13:20:56,923 INFO mapreduce.Job:  map 0% reduce 0%
2023-04-19 13:21:00,998 INFO mapreduce.Job:  map 100% reduce 0%
```

Step 9: Viewing output files

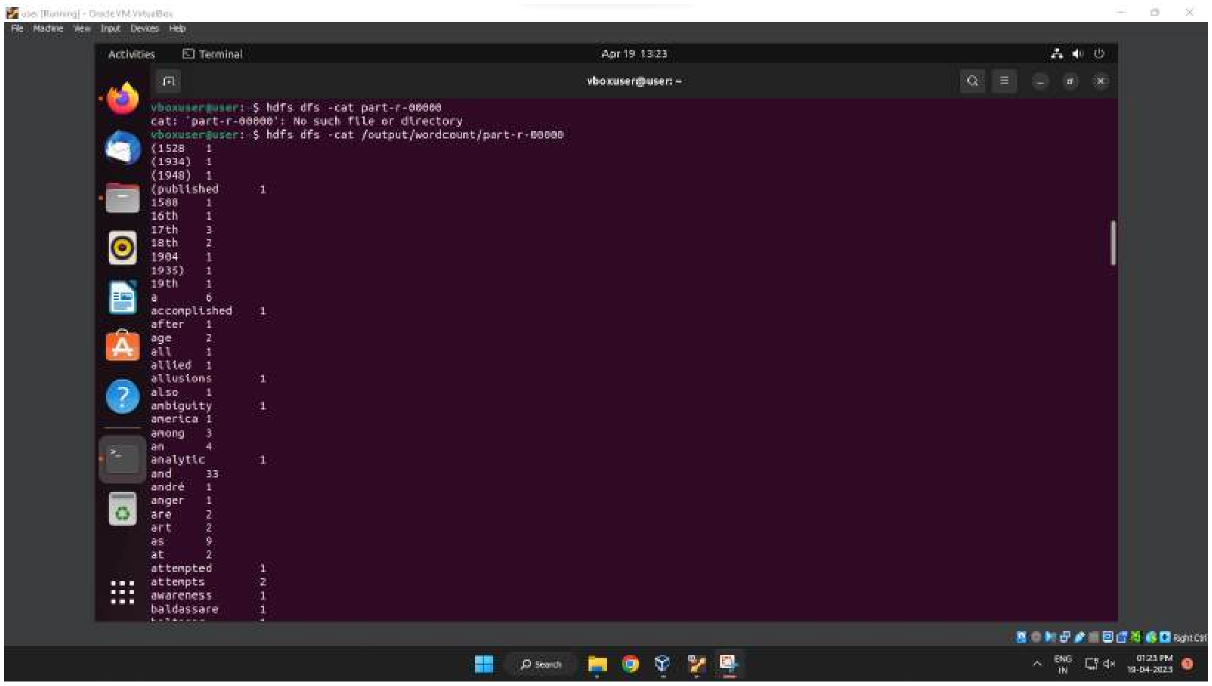
$ hdfs dfs –ls/output/wordcount



```
                GC time elapsed (ms)=156
                CPU time spent (ms)=1120
                Physical memory (bytes) snapshot=549984384
                Virtual memory (bytes) snapshot=5124988928
                Total committed heap usage (bytes)=425721856
                Peak Map Physical memory (bytes)=318214144
                Peak Map Virtual memory (bytes)=2558824448
                Peak Reduce Physical memory (bytes)=231698240
                Peak Reduce Virtual memory (bytes)=2566164488
        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=1580
        File Output Format Counters
                Bytes Written=2953
vboxuser@user:~$ hdfs dfs -ls /output/wordcount
Found 2 items
-rw-r--r--   1 vboxuser supergroup          0 2023-04-19 13:21 /output/wordcount/_SUCCESS
-rw-r--r--   1 vboxuser supergroup       2953 2023-04-19 13:21 /output/wordcount/part-r-00000
vboxuser@user:~$ hdfs dfs -cat part-r-00000
cat: 'part-r-00000': No such file or directory
vboxuser@user:~$ hdfs dfs -cat /output/wordcount/part-r-00000
(1528   1
(1934)  1
(1948)  1
(published      1
1588    1
16th    1
17th    3
18th    2
1904    1
1935)   1
19th    1
```

Step 10: Final output

$ hdfs dfs –cat /output/wordcount/part-r-00000



```
vboxuser@user:~$ hdfs dfs -cat part-r-00000
cat: 'part-r-00000': No such file or directory
vboxuser@user:~$ hdfs dfs -cat /output/wordcount/part-r-00000
(1528   1
(1934)  1
(1948)  1
(published      1
1588    1
16th    1
17th    3
18th    2
1904    1
1935)   1
19th    1
a       6
accomplished    1
after   1
age     2
all     1
allied  1
allusions       1
also    1
ambiguity       1
america 1
among   3
an      4
analytic        1
and     33
andré   1
anger   1
are     2
art     2
as      9
at      2
attempted       1
attempts        2
awareness       1
baldassare      1
```

**Result**

Thus, the implementation of word count/ frequency programs using MapReduce was executed.

## Ex.No.3 Implement a MR program that process a weather dataset

### Aim

To implement a MR program that process a weather dataset.

### Pre-lab Discussion

### Inputs and Outputs (Java Perspective)

The MapReduce framework operates on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.

The key and the value classes should be in serialized manner by the framework and hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable-Comparable interface to facilitate sorting by the framework. Input and Output types of a **MapReduce job** − (Input) <k1, v1> → map → <k2, v2> → reduce → <k3, v3>(Output).

| | Input | Output |
|---|---|---|
| **Map** | <k1, v1> | list (<k2, v2>) |
| **Reduce** | <k2, list(v2)> | list (<k3, v3>) |

### Terminology

- **PayLoad** − Applications implement the Map and the Reduce functions, and form the core of the job.
- **Mapper** − Mapper maps the input key/value pairs to a set of intermediate key/value pair.
- **NamedNode** − Node that manages the Hadoop Distributed File System (HDFS).
- **DataNode** − Node where data is presented in advance before any processing takes place.
- **MasterNode** − Node where JobTracker runs and which accepts job requests from clients.
- **SlaveNode** − Node where Map and Reduce program runs.
- **JobTracker** − Schedules jobs and tracks the assign jobs to Task tracker.
- **Task Tracker** − Tracks the task and reports status to JobTracker.
- **Job** − A program is an execution of a Mapper and Reducer across a dataset.
- **Task** − An execution of a Mapper or a Reducer on a slice of data.
- **Task Attempt** − A particular instance of an attempt to execute a task on a SlaveNode.

**Program**

**AverageMapper.java**

```
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*; import java.io.IOException;

public class AverageMapper extends Mapper <LongWritable, Text, Text, IntWritable>
{
public static final int MISSING = 9999;
public void map(LongWritable key, Text value, Context context) throwsIOException, InterruptedException
{
String line = value.toString();
String year = line.substring(15,19);
int temperature;
if (line.charAt(87)=='+')
temperature = Integer.parseInt(line.substring(88, 92));
else
temperature = Integer.parseInt(line.substring(87, 92));
String quality = line.substring(92, 93);
if(temperature != MISSING && quality.matches("[01459]")) context.write(new Text(year),new
IntWritable(temperature));
}
}
```

**AverageReducer.java**

```
import org.apache.hadoop.mapreduce.*;
import java.io.IOException;
public class AverageReducer extends Reducer <Text, IntWritable,Text, IntWritable >
{
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException
{
int max_temp = 0;
int count = 0;
for (IntWritable value : values)
{
max_temp += value.get();
count+=1;
}
context.write(key, new IntWritable(max_temp/count));
}        }
```

**AverageDriver.java**

```
import.org.apache.hadoop.io.*;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class AverageDriver
{
public static void main (String[] args) throws Exception
{
if (args.length != 2)
{
System.err.println("Please Enter the input and output parameters");
System.exit(-1);
}
Job job = new Job(); job.setJarByClass(AverageDriver.class); job.setJobName("Max temperature");

FileInputFormat.addInputPath(job,newPath(args[0])); FileOutputFormat.setOutputPath(job,new Path
(args[1]));

job.setMapperClass(AverageMapper.class); job.setReducerClass(AverageReducer.class);
job.setOutputKeyClass(Text.class); job.setOutputValueClass(IntWritable.class)
System.exit(job.waitForCompletion(true)?0:1);
}
}
```

**Result**

Thus, the implementation of an MR program that process a weather dataset was executed successfully.

## Ex.No.4 Implement Linear and Logistic Regression

## Aim

To implement linear and logistic regression for Sales dataset.

**Pre-lab Discussion**

**Theory**

**Linear Regression:**

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable. It consists of 3 stages –

(1) analyzing the correlation and directionality of the data

(2) estimating the model, i.e., fitting the line

(3) evaluating the validity and usefulness of the model.

The linear regression model provides a sloped straight line representing the relationship between the variables. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

**Types of Linear Regression**

Linear regression can be further divided into two types of the algorithm:

**SimpleLinearRegression**

If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

**Multiple Linear regression**

If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

The main function to calculate values of coefficients

Initialize the parameters.

Predict the value of a dependent variable by given an independent variable.

Calculate the error in prediction for all data points.

Calculate partial derivative w.r.t a0 and a1.

Calculate the cost for each number and add them.

Update the values of a0 and a1.

**Logistic Regression:**

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

**Type of Logistic Regression:**

On the basis of the categories, Logistic Regression can be classified into three types:

**Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

**Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"

**Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

**Applications of Logistic Regression**

1. Predicting a probability of a person having a heart attack
2. Predicting a customer’s propensity to purchase a product or halt a subscription.
3. Predicting the probability of failure of a given process or product.

**Steps in Logistic Regression:**

- Data Pre-processing step
- Fitting Logistic Regression to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.

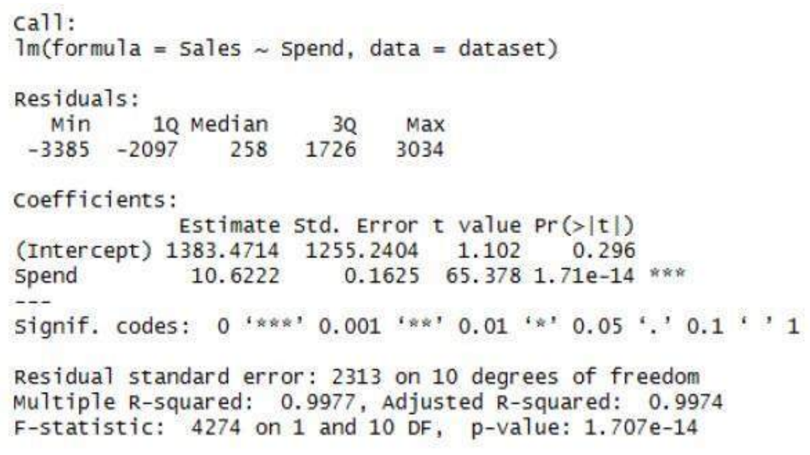
**PROGRAM:**

******SIMPLE LINEAR REGRESSION******

```
dataset = read.csv("data-marketing-budget-12mo.csv", header=T,
colClasses = c("numeric", "numeric", "numeric"))
head(dataset,5)

//////Simple Regression/////

simple.fit = lm(Sales~Spend,data=dataset)
summary(simple.fit)
```

**OUTPUT:**

**SIMPLE LINEAR REGRESSION:**



```
Call:
lm(formula = Sales ~ Spend, data = dataset)

Residuals:
   Min     1Q Median     3Q    Max
 -3385  -2097    258   1726   3034

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1383.4714  1255.2404   1.102    0.296
Spend         10.6222     0.1625  65.378 1.71e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2313 on 10 degrees of freedom
Multiple R-squared:  0.9977, Adjusted R-squared:  0.9974
F-statistic:  4274 on 1 and 10 DF,  p-value: 1.707e-14
```

**PROGRAM:**

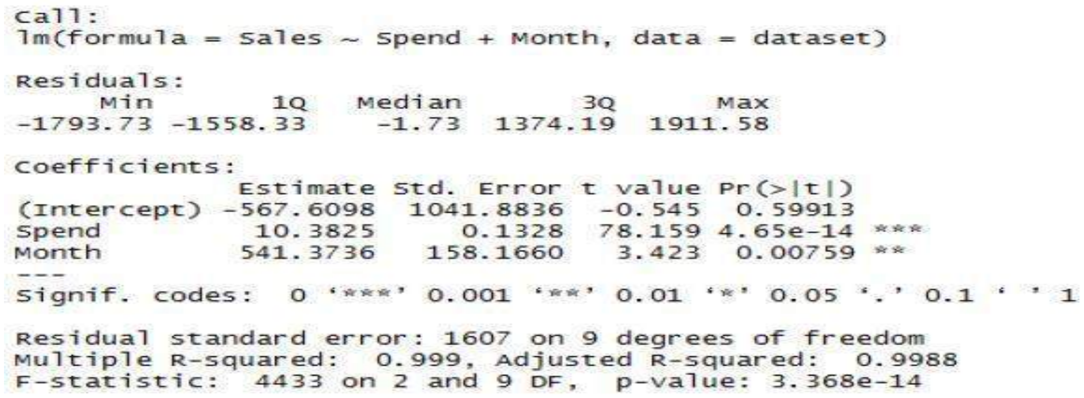
******MULTIPLE LINEAR REGRESSION ******

multi.fit = lm(Sales~Spend+Month, data=dataset)

summary(multi.fit)

**OUTPUT:**

**MULTIPLE LINEAR REGRESSION**



```
Call:
lm(formula = Sales ~ Spend + Month, data = dataset)

Residuals:
     Min       1Q   Median       3Q      Max
-1793.73 -1558.33    -1.73  1374.19  1911.58

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -567.6098  1041.8836  -0.545  0.59913
Spend         10.3825     0.1328  78.159 4.65e-14 ***
Month        541.3736   158.1660   3.423  0.00759 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1607 on 9 degrees of freedom
Multiple R-squared:  0.999,	Adjusted R-squared:  0.9988
F-statistic:  4433 on 2 and 9 DF,  p-value: 3.368e-14
```

**PROGRAM:**

******Logistic Regression ******

#selects some column from

mtcars input<- mtcars

[,c("am","cyl","hp","wt")]

print(head(input))

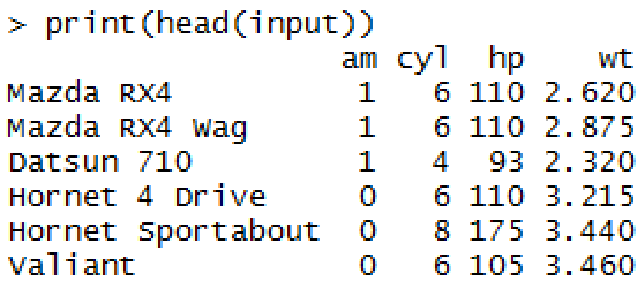
input<- mtcars [,c("am","cyl","hp","wt")]

am.data =glm(formula = am ~ cyl+hp+wt,data =

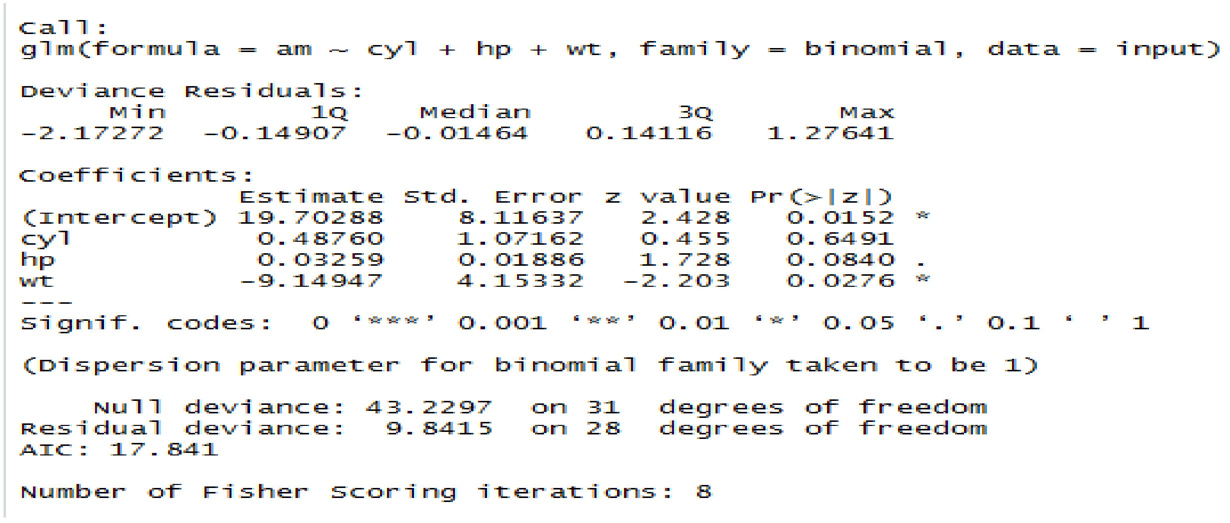
input,family = binomial)print(summary(am.data))

**OUTPUT:**

**LOGISTIC REGRESSION:**



```
> print(head(input))
                  am cyl  hp    wt
Mazda RX4          1   6 110 2.620
Mazda RX4 Wag      1   6 110 2.875
Datsun 710         1   4  93 2.320
Hornet 4 Drive     0   6 110 3.215
Hornet Sportabout  0   8 175 3.440
Valiant            0   6 105 3.460
```



```
Call:
glm(formula = am ~ cyl + hp + wt, family = binomial, data = input)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-2.17272  -0.14907  -0.01464   0.14116   1.27641  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)  
(Intercept) 19.70288    8.11637   2.428   0.0152 *
cyl          0.48760    1.07162   0.455   0.6491  
hp           0.03259    0.01886   1.728   0.0840 .
wt          -9.14947    4.15332  -2.203   0.0276 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.2297  on 31  degrees of freedom
Residual deviance:  9.8415  on 28  degrees of freedom
AIC: 17.841

Number of Fisher Scoring iterations: 8
```

## Result

Thus, the implementation of linear and logistic regression for sales data set was implemented successfully.

## Ex.No.5a Implement SVM Classification Techniques

## Aim

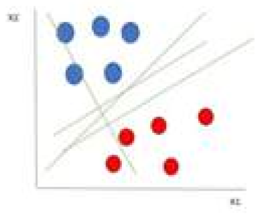
To implement support vector machine (SVM) to find optimum hyper plane Line in 2D, 3Dhyper plane) which maximize the margin between two classes.

**Pre-Lab Discussion**

**Theory**

**Support Vector Machine(SVM):**

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three. Let's consider two independent variables x1, x2, and one dependent variable which is either a blue circle or a red circle.



From the figure above it's very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregate our data points or do a classification between red and blue circles.

**Types of SVM**

SVM can be of two types:

- Linear SVM:

Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

- Non-linear SVM:

Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

**Hyperplane and Support Vectors in the SVM algorithm:**

**Hyperplane:** There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

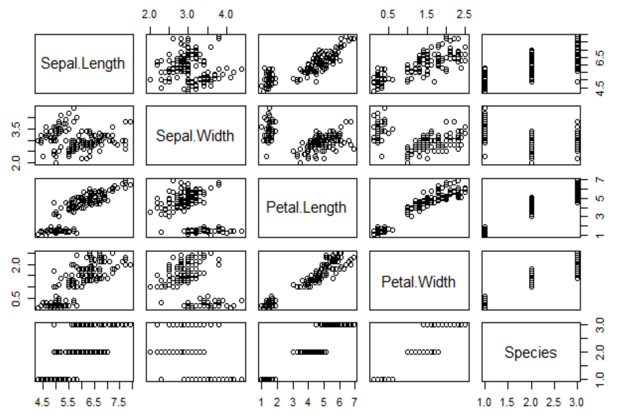
**Support Vectors:**

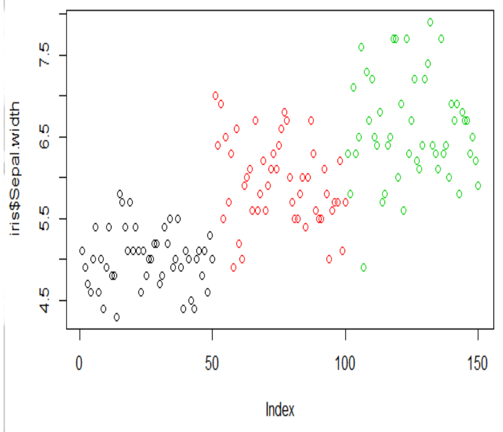
The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

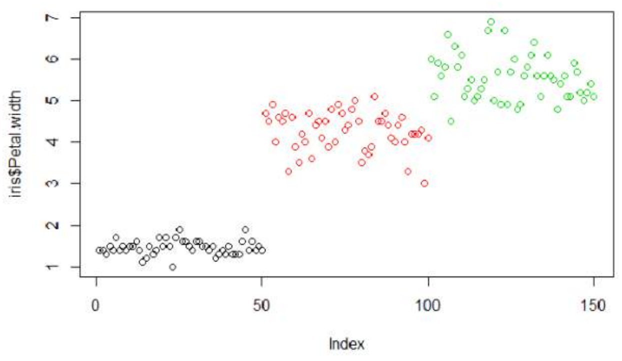
## PROGRAM:

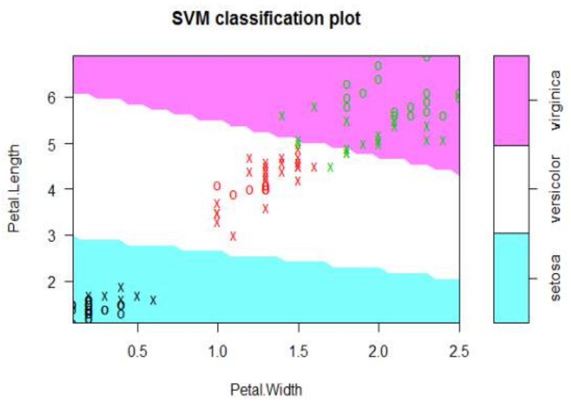
```
plot(iris)
iris
install.packages("e1071")
plot(iris$Sepal.Length, iris$Sepal.width, col=iris$Species)
plot(iris$Petal.Length, iris$Petal.width, col=iris$Species)
s<-sample(150,100)
col<- c("Petal.Length", "Petal.Width", "Species")
iris_train<- iris[s,col]
iris_test<- iris[-s,col]
svmfit<- svm(Species ~., data = iris_train, kernel = "linear", cost = .1, scale = FALSE)
print(svmfit)
plot(svmfit, iris_train[,col])
tuned <- tune(svm, Species~., data = iris_train, kernel = "linear", ranges=
list(cost=c(0.001,0.01,.1,.1,10,100)))
summary(tuned)
p<-predict(svmfit, iris_test[,col], type="class")
plot(p)
table(p,iris_test[,3] )
mean(p== iris_test[,3])
```

**OUTPUT:**









## Result

Thus , the implementation of support vector machine (SVM) to find optimum hyper plane (Line in 2D, 3Dhyper plane) which maximize the margin between two classes was executed.

## Ex.No.5b Implement Decision Tree Classification Techniques

### AIM

To implement a decision tree used to representing a decision situation in visually and show all those factors within the analysis that are considered relevant to the decision.

### Pre-Lab Discussion

### Theory

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.

### Steps:

Begin the tree with the root node, says S, which contains the complete dataset.

1. Find the best attribute in the dataset using Attribute Selection Measure (ASM).
2. Divide the S into subsets that contains possible values for the best attributes.
3. Generate the decision tree node, which contains the best attribute.
4. Recursively make new decision trees using the subsets of the dataset created in step - 3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

**Attribute Selection Measures**

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM. By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

- Information Gain
- Gini Index

**Information Gain:**

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- It calculates how much information a feature provides us about a class.
- According to the value of information gain, we split the node and build the decision tree.
- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first.

**Gini Index:**

- Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.

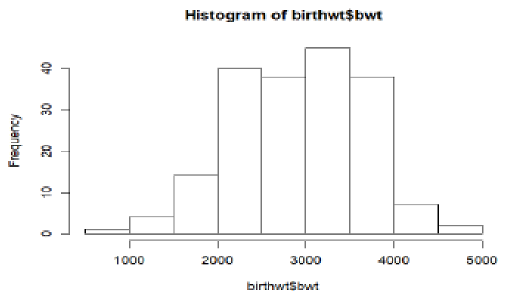
## PROGRAM:

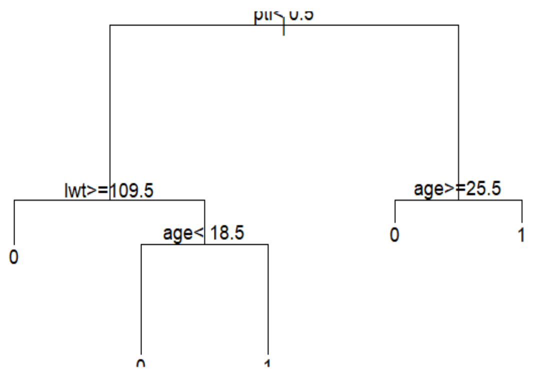
```
library(MASS)
library(rpart)
head(birthwt)
hist(birthwt$bwt)
table(birthwt$low)
cols <- c('low', 'race', 'smoke', 'ht', 'ui')
birthwt[cols] <- lapply(birthwt[cols], as.factor)
set.seed(1)
train<- sample(1:nrow(birthwt), 0.75 * nrow(birthwt))
birthwtTree<- rpart(low ~ . - bwt, data = birthwt[train, ], method = 'class')
```

```
plot(birthwtTree)
text(birthwtTree, pretty = 0)
summary(birthwtTree)
birthwtPred<- predict(birthwtTree, birthwt[-train, ], type = 'class')
table(birthwtPred, birthwt[-train, ]$low)
```

## OUTPUT:

```
    low   age  lwt  race  smoke ptl  ht  ui  ftv  bwt
85   0    19   182   2     0     0    0   1   0   2523
86   0    33   155   3     0     0    0   0   3   2551
87   0    20   105   1     1     0    0   0   1   2557
88   0    21   108   1     1     0    0   1   2   2594
89   0    18   107   1     1     0    0   1   0   2600
91    0   21   124   3     0     0    0   0   0   2622
```





## Result

Thus , the implementation of a decision tree used to representing a decision situation in visually and show allthose factors within the analysis that are considered relevant to the decision was executed.

## Ex.No.6 Implementation of Clustering Techniques

## Aim

To implement clustering techniques for iris data set.

**Pre-lab Discussion**

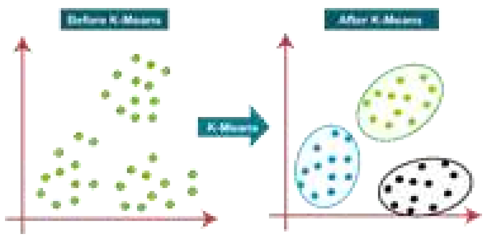
**Theory**

**K-Means Clustering**

K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm. The k-means clustering algorithm mainly performs two tasks:

1. Determines the best value for K center points or centroids by an iterative process.
2. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.



**Working of K-Means Algorithm:**

The working of the K-Means algorithm is explained in the below steps:

1. Select the number K to decide the number of clusters.

2. Select random K points or centroids. (It can be other from the input dataset).

3. Assign each data point to their closest centroid, which will form the predefined K clusters.

4. Calculate the variance and place a new centroid of each cluster.

5. Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

6. If any reassignment occurs, then go to step-4 else go to FINISH.

## PROGRAM:

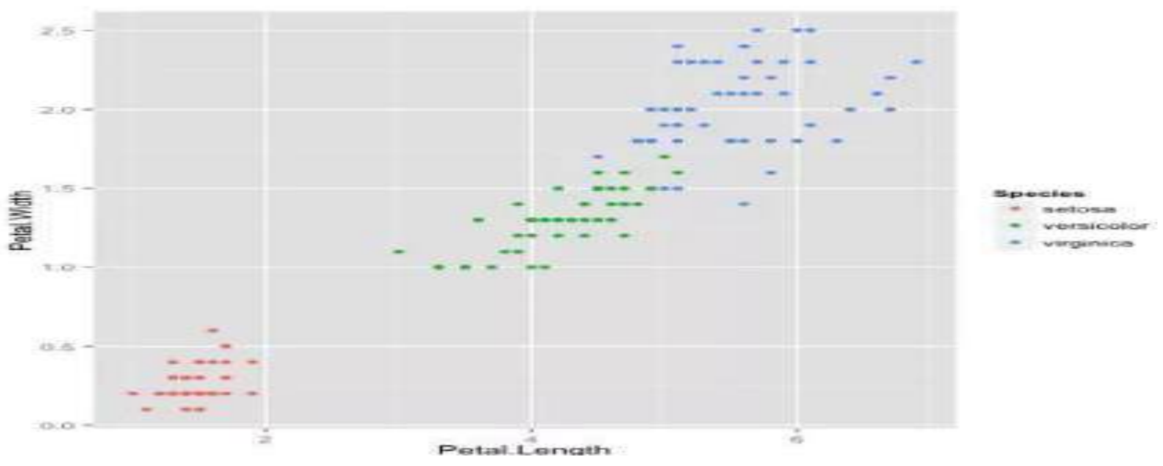
```
library(datasets)
head(iris)
library(ggplot2)
ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) + geom_point()
set.seed(20)
irisCluster <- kmeans(iris[, 3:4], 3, nstart = 20)
irisCluster
table(irisCluster$cluster, iris$Species)
```

## OUTPUT:

```
    Sepal.Length   Sepal.Width   Petal.Length   Petal.Width   Species
1        5.1           3.5            1.4           0.2        setosa
2        4.9           3.0            1.4           0.2        setosa
3        4.7           3.2            1.3           0.2        setosa
4        4.6           3.1            1.5           0.2        setosa
5        5.0           3.6            1.4           0.2        setosa
6        5.4           3.9            1.7          0.4         setosa
```

```
ggplot(iris, aes(Petal.Length, Petal.Width, color =
Species)) + geom_point()
```

Here is the plot:



```
irisCluster
K-means clustering with 3 clusters of sizes 46, 54, 50

Cluster means:
  Petal.Length Petal.Width
1     5.626087    2.047826
2     4.292593    1.359259
3     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3 3 3 3 3
 [35] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2
 [69] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 1 1
[103] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 2 2 1
1 1 1 1 1 1 1
[137] 1 1 2 1 1 1 1 1 1 1 1 1 1 1

Within cluster sum of squares by cluster:
[1] 15.16348 14.22741  2.02200
 (between_SS / total_SS =  94.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
```

## Result

Thus, K-means clustering using iris dataset was executed.

## Ex.No.7 Implementation of Visualize Data using Any Plotting Framework

**AIM**

To implement Data visualization is to provide an efficient graphical display for summarizingand reasoning about quantitative information.

**Pre-Lab Discussion**

**Theory**

### 1. Histogram

Histogram is basically a plot that breaks the data into bins (or breaks) and shows frequencydistribution of these bins. We can change the breaks also and see the effect it has data visualizationin terms of understandability.

### 2.1 Line Chart

The line chart showing the increase in air passengers over given time period. Line Chartsare commonly preferred when we are to analyses a trend spread over a time period. Furthermore,line plot is also suitable to plots where we need to compare relative changes in quantities across some variable (like time).

### 2.2 Bar Chart

Bar Plots are suitable for showing comparison between cumulative totals across several groups.Stacked Plots are used for bar plots for various categories.

### 3. Box Plot

Box Plot shows 5 statistically significant numbers the minimum, the 25th percentile, the median, the 75th percentile and the maximum. It is thus useful for visualizing the spread of the data is and deriving inferences accordingly.

### 4. Scatter Plot (including 3D and other features)

Scatter plots help in visualizing data easily and for simple data inspection.

### 5.Heat Map

One of the most innovative data visualizations in R, the heat map emphasizes color intensity to visualize relationships between multiple variables. The result is an attractive 2D imagethat is easy to interpret. As a basic example, a heat map highlights the popularity of competing items by ranking them according to their original market launch date. It breaks it down further byproviding sales statistics and figures over the course of time.

### 6.Correlogram

Correlated data is best visualized through corrplot. The 2D format is similar to a heat map, but it highlights statistics that are directly related. Most correlograms highlight the amount of correlation between datasets at various points in time.Comparing sales data between different months or years is a basic example.

### 7.Area Chart

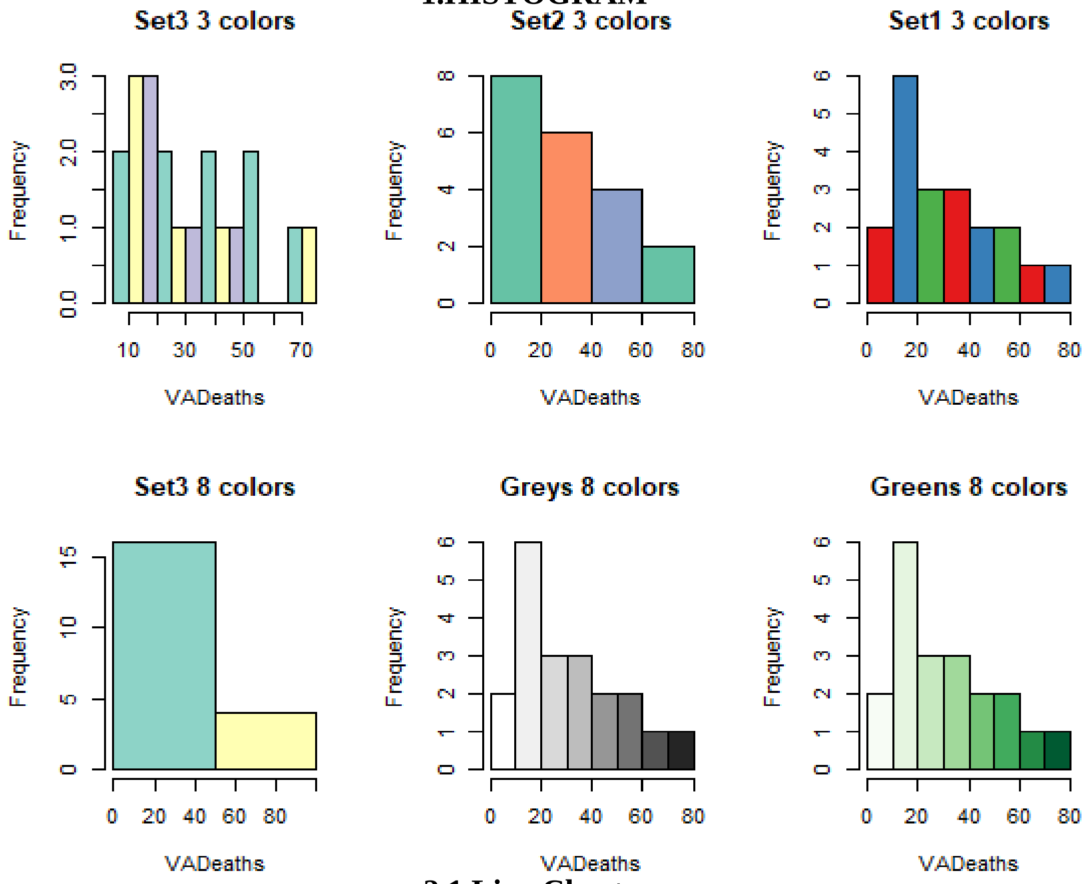
Area charts express continuity between different variables or data sets. It's akin to the traditionalline chart you know from grade school and is used in a similar fashion. Most area charts highlight trends and their evolution over the course of time, making them highly effective when trying to expose underlying trends whether they're positive or negative.

**PROGRAM:**

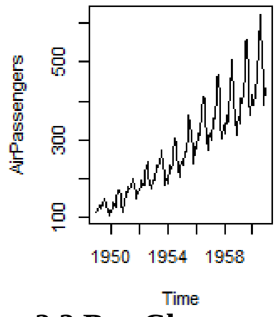
```
print('-----------------1.HISTOGRAM-------------------')
install.packages('RColorBrewer')
library(RColorBrewer)
data(VADeaths)
par(mfrow=c(2,3))
hist(VADeaths,breaks=10, col=brewer.pal(3,"Set3"),main="Set3 3 colors")
hist(VADeaths,breaks=3 ,col=brewer.pal(3,"Set2"),main="Set2 3 colors")
hist(VADeaths,breaks=7, col=brewer.pal(3,"Set1"),main="Set1 3 colors")
hist(VADeaths,,breaks= 2, col=brewer.pal(8,"Set3"),main="Set3 8 colors")
hist(VADeaths,col=brewer.pal(8,"Greys"),main="Greys 8 colors")
hist(VADeaths,col=brewer.pal(8,"Greens"),main="Greens 8 colors")
print('--------------- -2.1.        Line Chart-------------------')
data(AirPassengers)
plot(AirPassengers,type="l") #Simple Line Plot
print('-----------------2.2.        Bar Chart-------------------')
data("iris")
barplot(iris$Petal.Length) #Creating simple Bar Graph
barplot(iris$Sepal.Length,col = brewer.pal(3,"Set1"))
barplot(table(iris$Species,iris$Sepal.Length),col = brewer.pal(3,"Set1")) #Stacked Plot
print('-----------------3. Box Plot-------------------')
data(iris)
par(mfrow=c(2,2))
boxplot(iris$Sepal.Length,col="red")
boxplot(iris$Sepal.Length~iris$Species,col="red")
boxplot(iris$Sepal.Length~iris$Species,col=heat.colors(3))
boxplot(iris$Sepal.Length~iris$Species,col=topo.colors(3))
boxplot(iris$Petal.Length~iris$Species)
print('-----------------4.Scatter Plot -------------------')
plot(x=iris$Petal.Length) #Simple Scatter Plot
plot(x=iris$Petal.Length,y=iris$Species) #Multivariate Scatter Plot
print('-----------------5. Heat Map-------------------')
X<-rnorm(10,mean=rep(1:5,each=2),sd=0.7)
y<-rnorm(10,mean=rep(c(1,9),each=5),sd=0.1)
dataFrame<-data.frame(x=X,y=y)
set.seed(143)
dataMatrix<-as.matrix(dataFrame)[sample(1:10),] # convert to class 'matrix', then shuffle the
rows of the matrix
heatmap(dataMatrix) # visualize hierarchical clustering via a heatmap
print('-----------------6. Correlogram-------------------')
install.packages("corrplot")
#data("mtcars")
corr_matrix <- cor(mtcars)
# with circles
corrplot(corr_matrix)
# with numbers and lower
corrplot(corr_matrix,method = 'number',type = "lower")
print('-----------------    Area Chart-------------------')
install.packages("magrittr")
install.packages("dplyr") \
install.packages('tidyverse')
library(dplyr)
library(magrittr)
library(tidyverse)
data("airquality")
airquality %>% group_by(Day) %>%
summarise(mean_wind = mean(Wind)) %>%
ggplot() + geom_area(aes(x = Day, y = mean_wind)) + labs(title = "Area Chart of Average
Wind per Day",subtitle = "using airquality data", y = "Mean Wind")
```

**OUTPUT:**

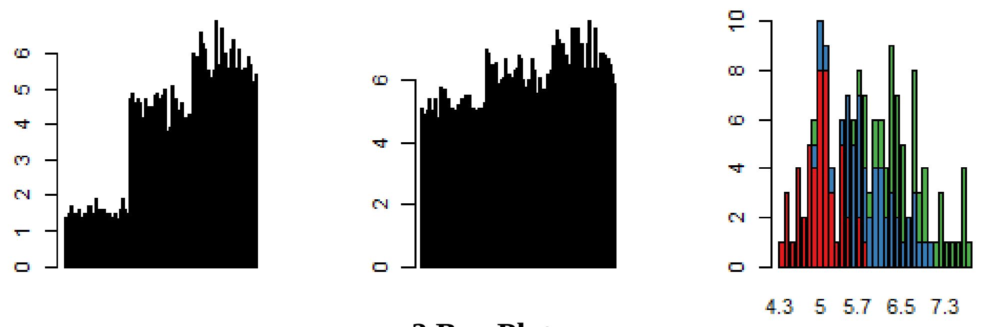
**-----------------1.HISTOGRAM------------------**



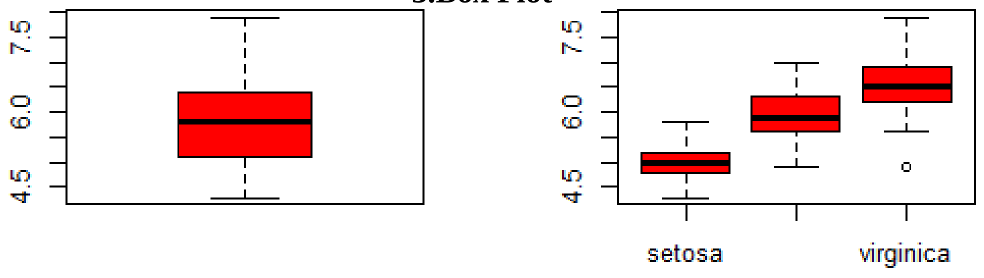
**---------------2.1.Line Chart------------------**

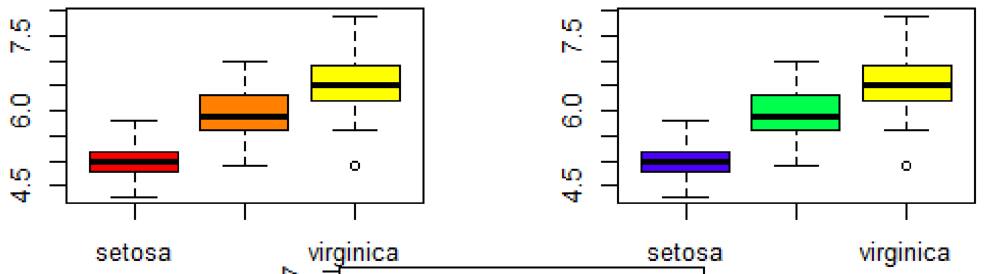


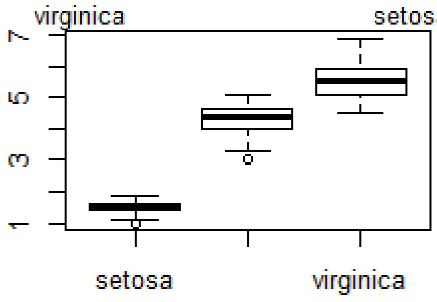
**----------------2.2.Bar Chart-----------------**



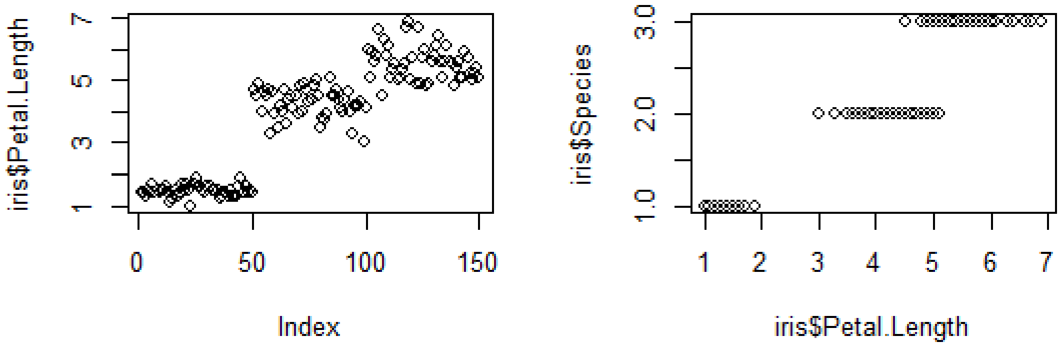
----------------3.Box Plot------------------



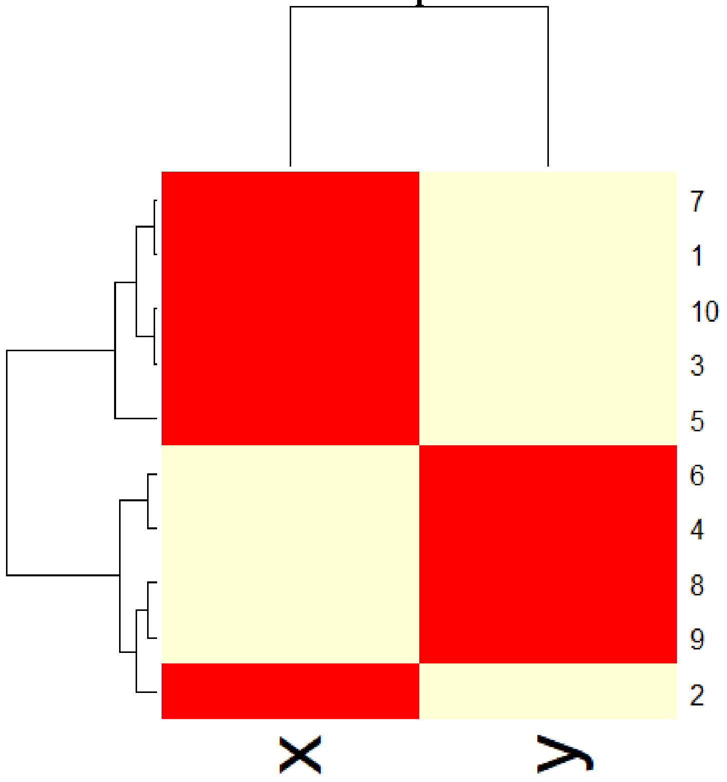




----------------4.Scatter Plot ------------------



-----------------**5.Heat Map**-------------------



-----------------**6.Correlogram**-------------------

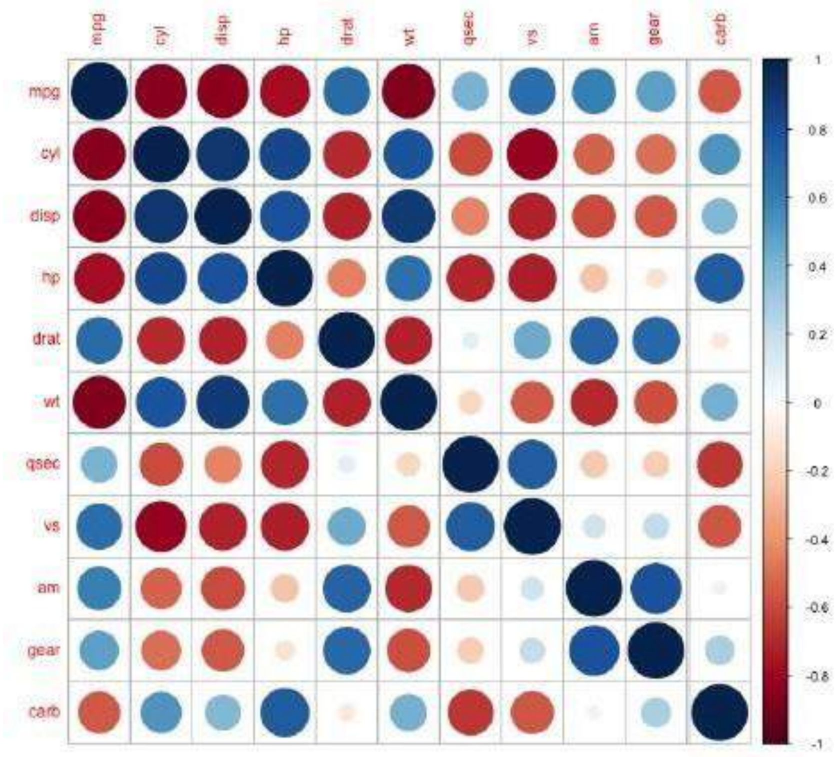


Fig 6. Correlogram with circles (courtesy of Abdul Majed Raja)

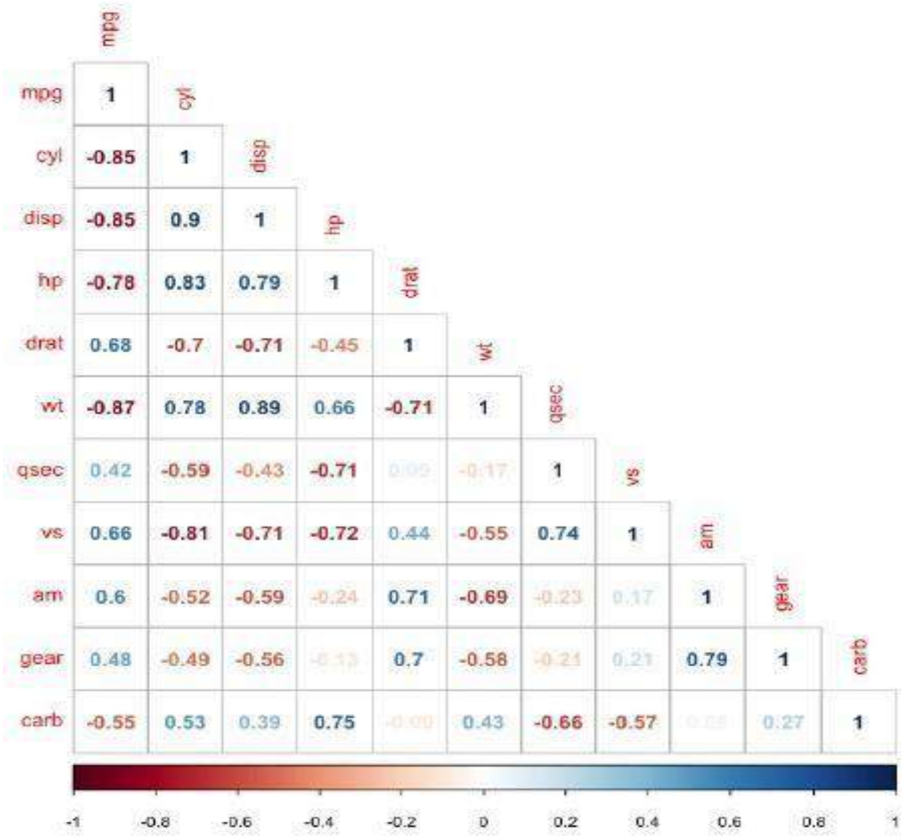


Fig 7. Correlogram with numbers (courtesy of Abdul Majed Raja)

**----------------- Area Chart-------------------**

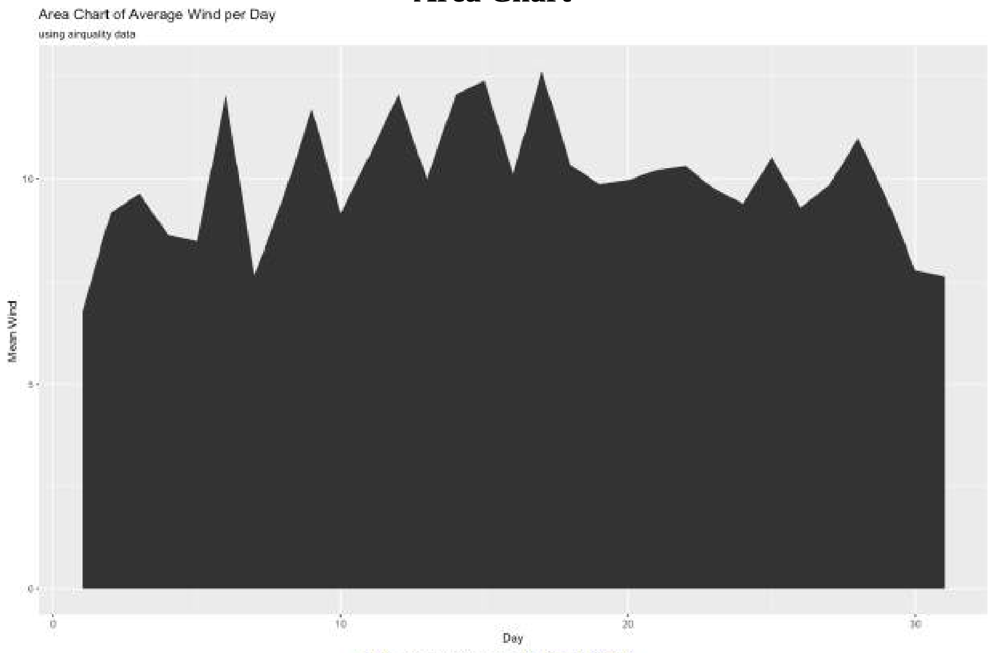


Fig 8. Area chart (courtesy of Abdul Majed Raja)

## Result

Thus, the implementation of data visualization was successfully executed.

## Ex.No.8 Implement an application that stores big data in Hbase / MangoDB/ Pig using Hadoop / R

## Aim

To implement an application that stores big data in MongoDB using R.

**Pre-Lab Discussion**

**Theory**

**MongoDB with R**

Mongodb is a NoSql database platform that works on the concept of collection and documents. Collection: Collections are just like tables in relational databases. They are a group of Mongodb documents. These collections contain a set of documents. Document: Documents are like tuples/ rows in a relational database. R provides several libraries for creating a connection between mongodb and R such as: mongolite, Rmongo, rmongodb .

Step 1 - Install 'RMango package'

```
install.packages("RMongo")
library(RMongo)
```

Step 2 - Create a connection

```
r_mongo_con <- mongoDbConnect('db')
```

Step 3 - Check the connection

print(dbShowCollections(r_mongo_con)) # this verifies the established connection , returns errors if any

Step 4 - Run Queries

```
var_Query <- dbGetQuery(mongo, 'collection_name', "{'type': 'required_data'}")
```

Step 5 - Install mongolite package

```
install.packages('mongolite') library(mongolite)
```

Step 6 - Create a connection

```
mongolite_conn <- mongo(dataset, url)
```

The most popular packages to connect MongoDB and R are:

**mongolite:** A more recent R MongoDB driver, mongolite can perform various operations like indexing, aggregation pipelines, TLS encryption, and SASL authentication, among others. It’s based on the jsonlite package for R and mongo-c-driver. We can install mongolite from CRAN or from RStudio (explained in a later section).

**RMongo:** RMongo was the first R MongoDB driver with a simple R MongoDB interface. It has syntax like the MongoDB shell. RMongo has been deprecated as of now.

rmongodb: rmongodb has functions to create pipelines, handle BSON objects, etc. Its syntax is very complex compared to mongolite. Just like RMongo, rmongodb has been deprecated and is not available or maintained on CRAN.

**Inserting data**

Let's insert the crimes data from data.gov to MongoDB. The dataset reflects reported incidents of crime (with the exception of murders where data exists for each victim) that occurred in the City of Chicago since 2001.

**PROGRAM:**

```
install.packages('gridExtra')
library (ggplot2)
library (dplyr)
library (maps)
library (ggmap)
library (mongolite)
library (lubridate)
library (gridExtra)
crimes=data.table::fread("crime.csv")
names (crimes)
names(crimes) = gsub(" ","",names(crimes))
names(crimes)
my_collection = mongo(collection = "crimes", db = "Chicago") # create connection,
database and collection
my_collection$insert(crimes)
my_collection$count()
my_collection$iterate()$one()
length(my_collection$distinct("PrimaryType"))
my_collection$count('{"PrimaryType":"ASSAULT" }')
query1= my_collection$find('{"PrimaryType" : "THEFT", "Domestic" : false }')
```

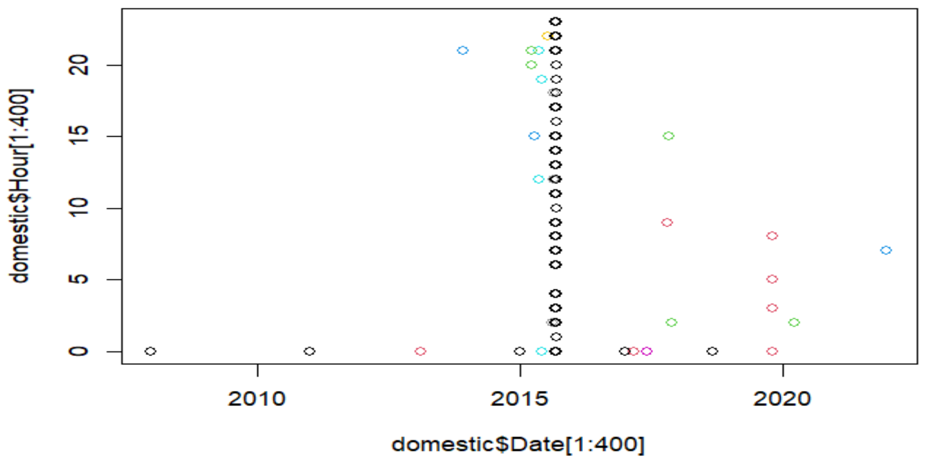
```
query2= my_collection$find('{"PrimaryType" : "THEFT", "Domestic" : true }',fields =
'{"_id":0, "PrimaryType":1, "Domestic":1}')
ncol(query1) # with all the columns
ncol(query2) # only the selected columns
domestic=my_collection$find('{"Domestic":true}',
fields   =        '{"_id":0}')
domestic$Date= mdy_hms(domestic$Date)
domestic$Weekday = weekdays(domestic$Date)
domestic$Hour = hour(domestic$Date)
domestic$month = month(domestic$Date,label=TRUE)
plot(domestic$Date,domestic$Hour, col=domestic$month)
pie(domestic)
barplot(domestic$Hour,domestic$month)
plot(domestic$District,domestic$Hour)
plot(domestic$District[1:1000], type="l", col="blue")
DayHourCounts = as.data.frame(table(domestic$Weekday, domestic$Hour))
DayHourCounts$Hour = as.numeric(as.character(DayHourCounts$Var2))
ggplot(DayHourCounts, aes(x=Hour, y=Freq)) + geom_line(aes(group=Var1, color=Var1),
size=1.4)+ylab("Count")+ ylab("Total Domestic Crimes")+ggtitle("Domestic Crimes in the
City of Chicago Since 2001")+theme(axis.title.x=element_text(size=14),axis.text.y =
element_text(color="blue",size=11,angle=0,hjust=1,vjust=0),axis.text.x
=element_text(color="blue",size=11,angle=0,hjust=.5,vjust=.5), axis.title.y =
element_text(size=14),legend.title=element_blank(),plot.title=element_text(size=16,color="
purple",hjust=0.5))
DayHourCounts$Type = ifelse((DayHourCounts$Var1 == "Sunday") |
(DayHourCounts$Var1 == "Saturday"), "Weekend", "Weekday")
ggplot(DayHourCounts, aes(x=Hour, y=Freq)) + geom_line(aes(group=Var1, color=Type),
size=2, alpha=0.5) +ylab("Total Domestic Crimes")+ggtitle("Domestic Crimes in the City of
Chicago Since 2001")+theme(axis.title.x=element_text(size=14),axis.text.y =
element_text(color="blue",size=11,angle=0,hjust=1,vjust=0),axis.text.x =
element_text(color="blue",size=11,angle=0,hjust=.5,vjust=.5), axis.title.y =
element_text(size=14),legend.title=element_blank(),plot.title=element_text(size=16,color="
purple",hjust=0.5))
DayHourCounts$Var1 = factor(DayHourCounts$Var1, ordered=TRUE,levels=c("Monday",
"Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
ggplot(DayHourCounts, aes(x = Hour, y = Var1)) + geom_tile(aes(fill = Freq)) +
scale_fill_gradient(name="Total MV Thefts", low="white", high="red") +ggtitle("Domestic
Crimes in the City of Chicago Since 2001")+theme(axis.title.y =
element_blank())+ylab("")+theme(axis.title.x=element_text(size=14),axis.text.y =
element_text(size=13),axis.text.x = element_text(size=13), axis.title.y =
element_text(size=14),legend.title=element_blank(),plot.title=element_text(size=16,color="
purple"))
domestic=my_collection$find('{"Domestic":true}',  fields  ='{"_id":0,
"Domestic":1,"Date":1}')
domestic$Date= mdy_hms(domestic$Date)
domestic$Weekday = weekdays(domestic$Date)
domestic$Hour = hour(domestic$Date)
domestic$month = month(domestic$Date,label=TRUE)
WeekdayCounts = as.data.frame(table(domestic$Weekday))
WeekdayCounts$Var1 = factor(WeekdayCounts$Var1, ordered=TRUE, levels=c("Sunday",
"Monday", "Tuesday", "Wednesday", "Thursday", "Friday","Saturday"))
ggplot(WeekdayCounts,aes(x=Var1, y=Freq))+geom_line(aes(group=1),size=2,color="red")
+ xlab("Day of the Week") + ylab("Total Domestic Crimes")+ ggtitle("Domestic Crimes in
the City of Chicago Since 2001")+ theme(axis.title.x=element_blank(),axis.text.y  =
element_text(color="blue",size=11,angle=0,hjust=1,vjust=0),axis.text.x =
element_text(color="blue",size=11,angle=0,hjust=.5,vjust=.5), axis.title.y =
element_text(size=14), plot.title=element_text(size=16,color="purple",hjust=0.5))
ASSAULT=my_collection$count('{"PrimaryType":"ASSAULT", "Domestic" : true }')
my_collection$aggregate('[{"$group":{"_id":"$LocationDescription","Count":{"$sum":1}}}
')%>%na.omit()%>%arrange(desc(count))%>%head(10)%>%
ggplot(aes(x=reorder(`_id`,count),y=count))+geom_bar(stat="identity",color='skyblue',fill='
#b35900')+geom_text(aes(label count), color = "blue") +coord_flip()+xlab("Location
Description")
```

**OUTPUT:**

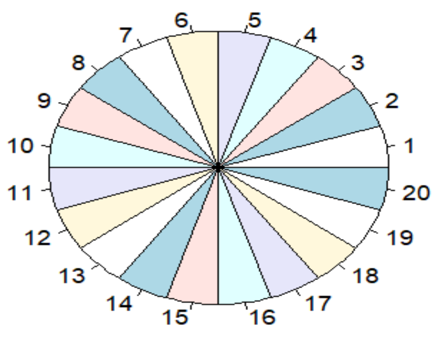
```
> names (crimes)
 [1] "ID"                   "Case Number"          "Date"
 [4] "Block"                "IUCR"                 "Primary Type"
 [7] "Description"          "Location Description" "Arrest"
[10] "Domestic"             "Beat"                 "District"
[13] "Ward"                 "Community Area"       "FBI Code"
[16] "X Coordinate"         "Y Coordinate"         "Year"
[19] "Updated On"           "Latitude"             "Longitude"
[22] "Location"
> names(crimes) = gsub(" ","",names(crimes))
> names(crimes)
 [1] "ID"                  "CaseNumber"          "Date"                "Block"
 [5] "IUCR"                "PrimaryType"         "Description"         "LocationDescription"
 [9] "Arrest"              "Domestic"            "Beat"                "District"
[13] "Ward"                "CommunityArea"       "FBICode"             "XCoordinate"
[17] "YCoordinate"         "Year"                "UpdatedOn"           "Latitude"
[21] "Longitude"           "Location"
> my_collection = mongo(collection = "crimes", db = "Chicago") # create connection,
database and collection
> my_collection$insert(crimes)
List of 5
 $ nInserted  : num 7750924
 $ nMatched   : num 0
 $ nRemoved   : num 0
 $ nUpserted  : num 0
 $ writeErrors: list()
> my_collection$count()
[1] 7750924
> my_collection$iterate()$one()
$ID
[1] 10224738

$CaseNumber
[1] "HY411648"

$Date
[1] "09/05/2015 01:30:00 PM"

$Block
[1] "043XX S WOOD ST"

$IUCR
[1] "0486"

$PrimaryType
[1] "BATTERY"

$Description
[1] "DOMESTIC BATTERY SIMPLE"
```

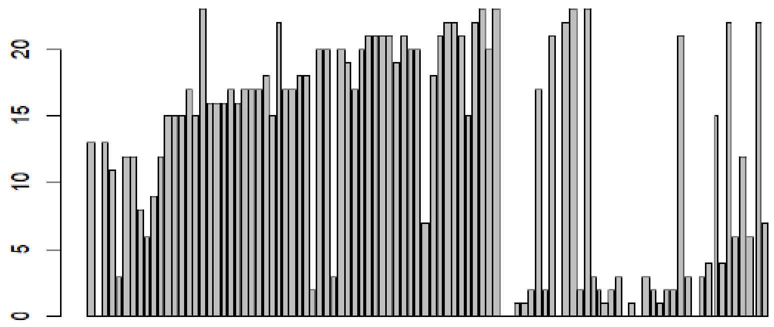
```
$LocationDescription
[1] "RESIDENCE"

$Arrest
[1] FALSE

$Domestic
[1] TRUE

$Beat
[1] 924

$District
[1] 9

$Ward
[1] 12

$CommunityArea
[1] 61

$FBICode
[1] "08B"

$XCoordinate
[1] 1165074

$YCoordinate
[1] 1875917

$Year
[1] 2015

$UpdatedOn
[1] "02/10/2018 03:50:01 PM"

$Latitude
[1] 41.81512

$Longitude
[1] -87.67

$Location
[1] "(41.815117282, -87.669999562)"

> length(my_collection$distinct("PrimaryType"))
[1] 36
> my_collection$count('{"PrimaryType":"ASSAULT" }')
[1] 504447
```

```
> query1= my_collection$find('{"PrimaryType" : "THEFT", "Domestic" : false }')
> query2= my_collection$find('{"PrimaryType" : "THEFT", "Domestic" : true }',
+                        fields = '{"_id":0, "PrimaryType":1, "Domestic":1}')
> ncol(query1) # with all the columns
[1] 22
> ncol(query2) # only the selected columns
[1] 2
> domestic=my_collection$find('{"Domestic":true}',
+ fields          =         '{"_id":0}')
> domestic$Date= mdy_hms(domestic$Date)
> domestic$Weekday = weekdays(domestic$Date)
> domestic$Hour = hour(domestic$Date)
> domestic$month = month(domestic$Date,label=TRUE)

> domestic$month = month(domestic$Date)
> plot(domestic$Date[1:400],domestic$Hour[1:400], col=domestic$month)
```

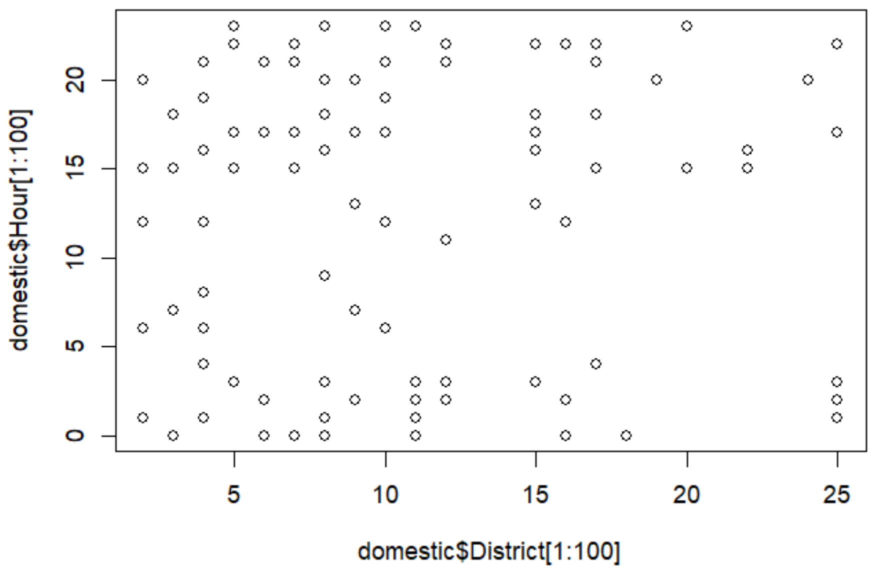


```
pie(domestic$Year[1:20])
```

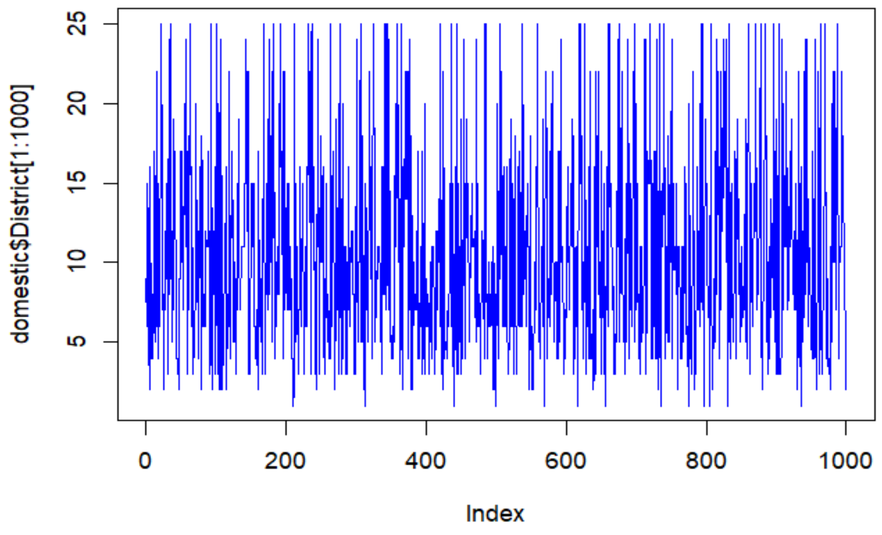




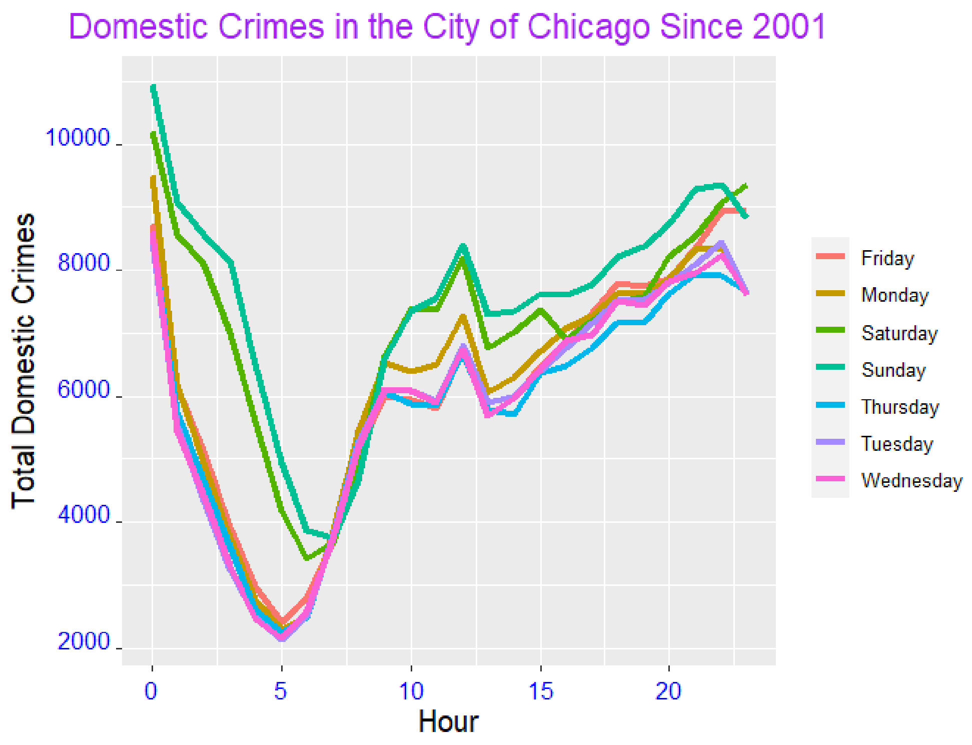
0
5
10
15
20

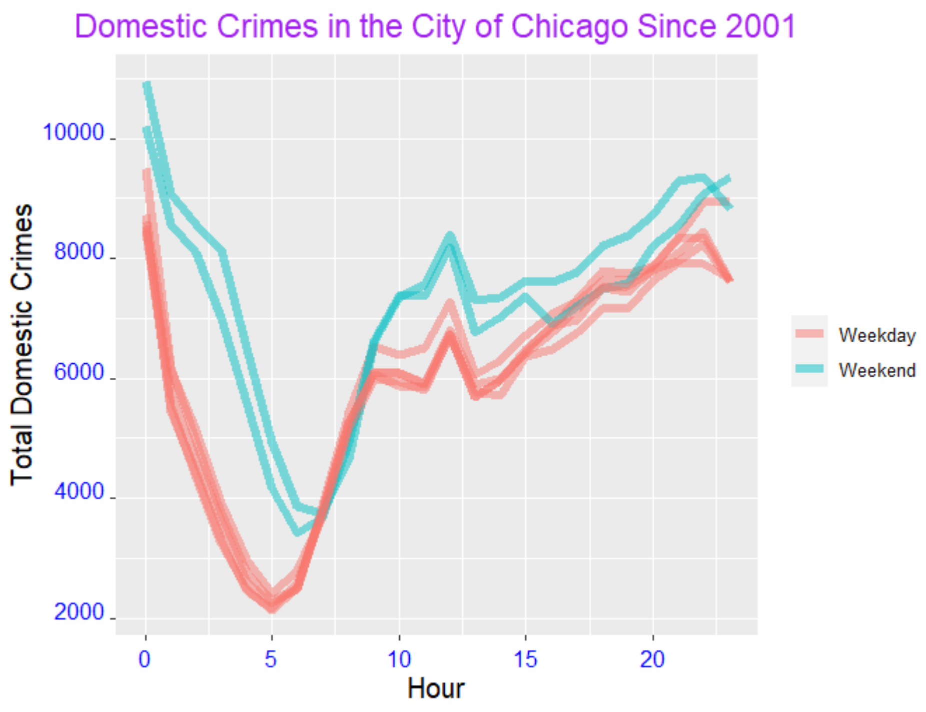


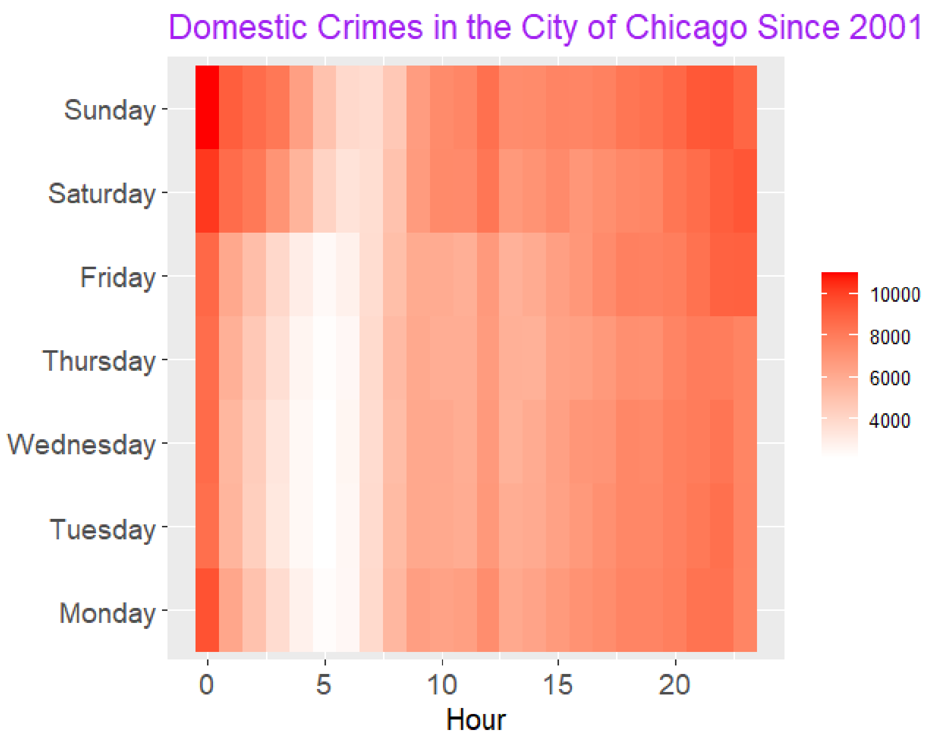
domestic$Hour[1:100]
domestic$District[1:100]
0
5
10
15
20
25

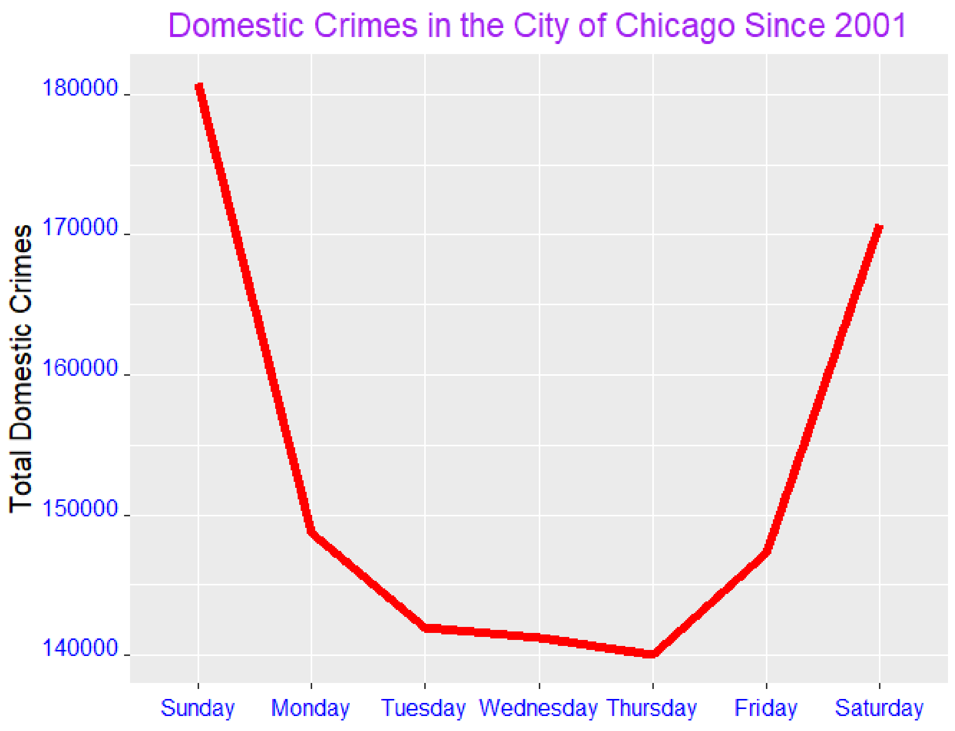


domestic$District[1:1000]
Index
5
10
15
20
25
0
200
400
600
800
1000


Domestic Crimes in the City of Chicago Since 2001
Total Domestic Crimes
10000
8000
6000
4000
2000
0
5
10
15
20
Hour
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday


Domestic Crimes in the City of Chicago Since 2001
Total Domestic Crimes
10000
8000
6000
4000
2000
0
5
10
15
20
Hour
Weekday
Weekend





**Result**

Thus, the application of Crime data set that stores big data in MongoDB using R was executed successfully.

**Exp.No.9** **CONTENT BEYOND SYLLABUS**

## RANDOM FOREST APPROACH FOR CLASSIFICATION

**Aim**

To implement Radom Forest approach for classification for iris data set.

**Theory**

Random forest approach is supervised nonlinear classification and regression algorithm. Classification is a process of classifying a group of datasets in categories or classes. As random forest approach can use classification or regression techniques depending upon the user and target or categories needed. A random forest is a collection of decision trees that specifies the categories with much higher probability. Random forest approach is used over decision trees approach as decision trees lack accuracy and decision trees also show low accuracy during the testing phase due to the process called over-fitting. In R programming, **randomForest()** function of **randomForest** package is used to create and analyze the random forest.

**Program:**

```
# Loading data
data(iris)
# Structure
str(iris)
#Installing package
install.packages("caTools")       # For sampling the dataset
install.packages("randomForest")  # For implementing random forest algorithm
# Loading package
library(caTools)
library(randomForest)
# Splitting data in train and test data
split <- sample.split(iris, SplitRatio = 0.7)
train <- subset(iris, split == "TRUE")
test <- subset(iris, split == "FALSE")
# Fitting Random Forest to the train dataset
set.seed(120)  # Setting seed
classifier_RF = randomForest(x = train[-5],
                             y = train$Species, ntree = 500)
classifier_RF
# Predicting the Test set results
y_pred = predict(classifier_RF, newdata = test[-5])
# Confusion Matrix
```

```
confusion_mtx = table(test[, 5], y_pred)

confusion_mtx
```
**# Plotting model**

```
plot(classifier_RF)
```

**# Importance plot**

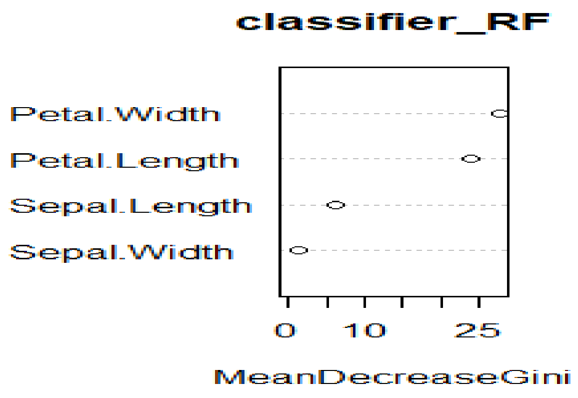
```
importance(classifier_RF)
```

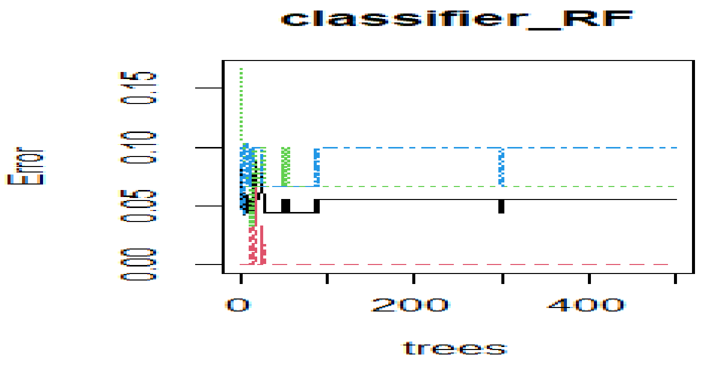
**# Variable importance plot**

```
varImpPlot(classifier_RF)
```

**OUTPUT:**

```
'data.frame':  150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
[1]  TRUE  TRUE FALSE  TRUE FALSE
Call:
 randomForest(x = train[-5], y = train$Species, ntree = 500)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2
  OOB estimate of  error rate: 5.56%
Confusion matrix:
                  Setosa     versicolor    virginica     class.error

setosa          30          0              0                0.00000000

versicolor       0        28               2             0.06666667

virginica        0          3             27                0.10000000

        y_pred

                    setosa     versicolor    virginica

  setosa           20            0           0

  versicolor        0           19           1

  virginica          0            2          18

                             MeanDecreaseGini

Sepal.Length          6.201739

Sepal.Width           1.527756

Petal.Length        23.936397

Petal.Width         27.591441
```





## Result

Thus, the implementation of random forest approach for classification was executed.

## K-Nearest Neighbor Classifier

**Aim**

To implement K-Nearest Neighbor Classifier for iris data set.

**Theory**

K-Nearest Neighbor or K-NN is a Supervised Non-linear classification algorithm. K-NN is a Non-parametric algorithm i.e it doesn’t make any assumption about underlying data or its distribution. It is one of the simplest and widely used algorithm which depends on it’s k value(Neighbors) and finds it’s applications in many industries like finance industry, healthcare industry etc.

In the KNN algorithm, K specifies the number of neighbors and its algorithm is as follows:

- Choose the number K of neighbor.
- Take the K Nearest Neighbor of unknown data point according to distance.
- Among the K-neighbors, Count the number of data points in each category.
- Assign the new data point to a category, where you counted the most neighbors.

For the Nearest Neighbor classifier, the distance between two points is expressed in the form of **Euclidean Distance.**

**Program**

```
# Loading data
data(iris)
# Structure
str(iris)
# Installing Packages
install.packages("e1071")
install.packages("caTools")
install.packages("class")
# Loading package
library(e1071)
library(caTools)
library(class)
# Loading data
data(iris)
head(iris)
# Splitting data into train and test data
split <- sample.split(iris, SplitRatio = 0.7)
```

```
train_cl <- subset(iris, split == "TRUE")
test_cl <- subset(iris, split == "FALSE")
# Feature Scaling
train_scale <- scale(train_cl[, 1:4])
test_scale <- scale(test_cl[, 1:4])
# Fitting KNN Model  to training dataset
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = train_cl$Species,
                      k = 1)
classifier_knn

# Confusiin Matrix
cm <- table(test_cl$Species, classifier_knn)
cm
# Model Evaluation - Choosing K
# Calculate out of Sample error
misClassError <- mean(classifier_knn != test_cl$Species)
print(paste('Accuracy =', 1-misClassError))
# K = 3
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = train_cl$Species,
                      k = 3)
misClassError <- mean(classifier_knn != test_cl$Species)
print(paste('Accuracy =', 1-misClassError))
# K = 5
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = train_cl$Species,
                      k = 5)
misClassError <- mean(classifier_knn != test_cl$Species)
print(paste('Accuracy =', 1-misClassError))
# K = 7
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = train_cl$Species,
                      k = 7)
misClassError <- mean(classifier_knn != test_cl$Species)
print(paste('Accuracy =', 1-misClassError))
# K = 15
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = train_cl$Species,
                      k = 15)
misClassError <- mean(classifier_knn != test_cl$Species)
print(paste('Accuracy =', 1-misClassError))
# K = 19
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                    cl = train_cl$Species,
                    k = 19)
misClassError <- mean(classifier_knn != test_cl$Species)
print(paste('Accuracy =', 1-misClassError))
```

**OUTPUT:**

```
'data.frame':  150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

   Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
1       5.1          3.5          1.4           0.2
2       4.9          3.0          1.4           0.2
3       4.7          3.2          1.3           0.2
4       4.6          3.1          1.5           0.2
5       5.0          3.6          1.4           0.2
6       5.4          3.9          1.7           0.4
    Species
1    setosa
2    setosa
3   setosa
4   setosa
5   setosa
6   setosa

[1] setosa     setosa     setosa     setosa
 [5] setosa     setosa     setosa     setosa
 [9] setosa     setosa     setosa     setosa
[13] setosa     setosa     setosa     setosa
[17] setosa     setosa     setosa     setosa
[21] versicolor versicolor versicolor versicolor
[25] versicolor versicolor versicolor versicolor
[29] virginica  versicolor versicolor versicolor
[33] versicolor virginica  versicolor versicolor
[37] versicolor versicolor versicolor versicolor
[41] virginica  virginica  virginica  virginica
[45] virginica  virginica  virginica  virginica
[49] virginica  virginica  virginica  virginica
[53] virginica  versicolor virginica  virginica
[57] virginica  virginica  virginica  virginica
Levels: setosa versicolor virginica

      classifier_knn
             setosa   versicolor   virginica
  setosa        20         0            0
  versicolor     0        18            2
  virginica      0         1           19
```

```
[1] "Accuracy = 0.95"
[1] "Accuracy = 0.95"
[1] "Accuracy = 0.966666666666667"
[1] "Accuracy = 0.983333333333333"
[1] "Accuracy = 0.966666666666667"
```

## Result

Thus, the K-NN classifier using iris data set was executed.

## Naive Bayes Classifier

### Aim

To implement Navie Bayes Classifier for iris dataset.

### Theory

Naive Bayes is a Supervised Non-linear classification algorithm in R Programming. Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Baye's theorem with strong(Naive) independence assumptions between the features or variables. The Naive Bayes algorithm is called "Naive" because it makes the assumption that the occurrence of a certain feature is independent of the occurrence of other features.

Naive Bayes algorithm is based on Bayes theorem. Bayes theorem gives the conditional probability of an event A given another event B has occurred.

$$P(A|B) = \frac{P(B|A)\, P(A)}{P(B)}$$

**where,**
P(A|B) = Conditional probability of A given B.
P(B|A) = Conditional probability of B given A.
P(A) = Probability of event A.
P(B) = Probability of event B.

### Program

```
# Loading data
data(iris)
# Structure
str(iris)
# Installing Packages
install.packages("e1071")
install.packages("caTools")
install.packages("caret")
# Loading package
library(e1071)
library(caTools)
library(caret)
# Splitting data into train and test data
split <- sample.split(iris, SplitRatio = 0.7)
train_cl <- subset(iris, split == "TRUE")
test_cl <- subset(iris, split == "FALSE")
# Feature Scaling
train_scale <- scale(train_cl[, 1:4])
test_scale <- scale(test_cl[, 1:4])
# Fitting Naive Bayes Model  to training dataset
set.seed(120)  # Setting Seed
classifier_cl <- naiveBayes(Species ~ ., data = train_cl)
classifier_cl
# Predicting on test data'
y_pred <- predict(classifier_cl, newdata = test_cl)
# Confusion Matrix
cm <- table(test_cl$Species, y_pred)
cm
# Model Evaluation
confusionMatrix(cm)
```

**OUTPUT:**

```
'data.frame':  150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

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      4.943333    0.3766306
  versicolor 6.000000    0.5051459
  virginica   6.500000    0.6817827
          Sepal.Width
    Y                  [,1]            [,2]
  setosa       3.400000   0.3859605
  versicolor  2.746667   0.3104317
```

```
virginica    2.926667   0.3362402
      Petal.Length
   Y            [,1]        [,2]
setosa     1.426667    0.1552158
versicolor   4.306667   0.5172429
virginica    5.486667   0.6179685

          Petal.Width
    Y            [,1]       [,2]
setosa     0.250000    0.1196259
versicolor 1.330000   0.2019730
virginica  1.976667   0.2160513
y_pred
             setosa   versicolor   virginica
setosa          20         0              0
versicolor       0        19              1
virginica        0         2             18
Confusion Matrix and Statistics
y_pred
           setosa   versicolor   virginica
setosa        20         0             0
versicolor     0        19             1
virginica      0         2             18

Overall Statistic:
               Accuracy : 0.95
                 95% CI : (0.8608, 0.9896)
    No Information Rate : 0.35
    P-Value [Acc > NIR] : < 2.2e-16
                  Kappa : 0.925
 Mcnemar's Test P-Value : NA

Statistics by Class:
                     Class: setosa        Class: versicolor
Sensitivity                1.0000          0.9048
Specificity                1.0000          0.9744
Pos Pred Value             1.0000          0.9500
Neg Pred Value             1.0000          0.9500
Prevalence                 0.3333          0.3500
Detection Rate             0.3333          0.3167
Detection Prevalence          0.3333          0.3333
Balanced Accuracy             1.0000          0.9396
                                   Class: virginica
Sensitivity                         0.9474
Specificity                         0.9512
Pos Pred Value                      0.9000
Neg Pred Value                      0.9750
Prevalence                          0.3167
Detection Rate                      0.3000
Detection Prevalence                0.3333
```

| | |
|---|---|
| **Balanced Accuracy** | 0 |

**Result**

Thus, the implementation of Navie Bayes theorem for iris data set was executed.

## VIVA Questions

1. What is Hadoop?
2. What platform and java version are required to run Hadoop?
3. What kind of hardware is best for Hadoop?
4. What are the most common input formats defined in Hadoop?
5. Explain the use of .mecia class?
6. Give the use of the bootstrap panel.
7. What is job tracker in Hadoop?
8. What are the difference between regular file system and HDFS?
9. Define name node.
10. Define data node.
11. Define mapreduce.
12. How does rack awareness work in HDFS?
13. How can you restart name node and the deamons in Hadoop?
14. Which command will help you to find the status of blocks and file system health?
15. How do you copy data from the local system onto HDFS?
16. Is logistic regression a generative or a descriptive classifier?
17. Can you use logistic regression for classification between more than two classes?
18. How do you implement multinomial logistic regression?
19. Why can't we use the mean square error cost function used in linear regression for logistic regression?
20. What alternative could you suggest using a for loop (which is time-consuming) when using Gradient Descent to find the optimum parameters for logistic regression?
21. Are there alternatives to find optimum parameters for logistic regression besides using Gradient Descent?
22. How many binary classifiers would you need to implement one-vs-one for four classes? How does it work?
23. What is the importance of regularisation?
24. When Logistic Regression can be used?
25. Why is Logistic Regression called Regression and not Classification?
26. What is linear regression?
27. Explain L1 and L2 regularisations.

28. Your linear regression doesn’t run and communicates that there is an infinite number of best estimates for the regression coefficients. What could be wrong?
29. How do you know that linear regression is suitable for any given data?
30. How is hypothesis testing used in linear regression?
31. Explain gradient descent with respect to linear regression.
32. What is the generalized linear model?
33. Which graphs are suggested to be observed before model fitting?
34. What is heteroscedasticity?

35. What are the possible ways of improving the accuracy of a linear regression model?

36. What are Decision Trees?
37. Explain the structure of a Decision Tree.
38. What are some advantages of using Decision Trees?
39. What is Gini Index and how is it used in Decision Trees?
40. What is Greedy Splitting?
41. What is Tree Boosting?
42. Why do you need to Prune the decision tree?
43. List down some popular algorithms used for deriving Decision Trees along with their attribute selection measures.
44. How are the different nodes of decision trees represented?
45. What is Support Vector Machine?
46. Name some advantages of SVM
47. What are Support Vectors in SVMs?
48. Compare SVM and Logistic Regression in handling outliers
49. What are Polynomial Kernels?
50. When SVM is not a good approach?
51. While designing an SVM classifier, what values should the designer select?
52. Why do we need to use Support Vector Machines?
53. What are Hard Margin SVMs and Soft Margin SVMs?
54. What is the relationship between Slack and Margin?