```
hadoop@DESKTOP-24S8RFC:$ sudo adduser --ingroup hadoop vishaka
Adding user `vishakha' ----
Adding new `vishakha' (1001) with group `hadoop' ---
Adding new `vishakha' (1001) with group `hadoop' ---
Creating home directory `/home/vishaka' ---
Enter the new value or press ENTER for the default
  Full Name [ ]: viva
  Room Number [ ]:
  Work Phone [ ]:
  Home Phone [ ]:
  Other [ ]:
Is the information correct ? [Y/n] Y
hadoop@DESKTOP-24S8RFC:$
```

```
hadoop@DESKTOP-84S8RFC:$ Sudo usermod -aG sudo
hadoop@DESKTOP-24S8RFC:$ sudo vishakha.
Password
To run a command as administrator (user "root"),
use "sudo <command>",
See "man sudo_root" for details.
```

```
Generating public/private rsa key pair
The key fingerprint is
The keys randomart image is :
+---- [RSA 3072] ---+
|                   |
|        . - 0      |
|     0 - - + +     |
|   B.  + + =       |
|   + x - - = s     |
|   = . + B + = + = |
|   0 . -- x+ + + 0 |
|   . .  =.E0 " + 0 0.|
+----[SHA256]-----+
```

Practical 1

Aim: Install, configure and run Hadoop and HDFS.

Start TERMINAL and enter below commands to check version of java and Hadoop.

Command 1: Sudo addgroup hadoop
Command 2: Sudo adduser - ingroup hadoop Vishakha

Command 3: Sudo usermod - aG sudo Vishakha
Command 4: su Vishakha

Command 5: ssh-keygen -t rsa - P ""

```
vishakha@DESKTOP-2458RFC:/home/hadoop$ cat $HOME/.ssh/id_rsa.
                                        pub >>$HOME/.ssh
vishakha@DESKTOP-2458RFC:/home/hadoop$ sudo nano/etc/sysctl.conf
[sudo] password for vishakha:
```

```
# for what other value do
# kernel.sysrq = 438
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv = 1

^G Help     ^O Write out   ^W Where Is    ^K Cut
^X Exit     ^R Read File   ^\ Replace     ^U Paste
```

```
Resolving mirrors.estointernet.in (mirrors.estointernet)
Connecting to mirrors.estointeinet.in
HTTP request sent, awaiting response 200 OK
Length 605187279 (577m) [application/octet-statement]
Saving to: 'hadoop-3.3.1 tar.gz
hadoop - 3.3.1 tar.gz
```

```
total 591044
drwxr-xr-x 2 root root 4096 NOV 23 03:06
drwxr-xr-x 2 root root 4096 NOV 23 03:06
drwxr-xr-x 2 root root 4096 NOV 23 03:6
-rw-r--r-- 1 vishakha hadoop 625187279 JUN 15 2021
```

DATE:

Command 6: Cat $HOME/.ssh/id-rsa.pub >> $HOME/.ssh/
authorized_keys
Command 7: Sudo nano/etc/sysctl.Conf

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1

Command 8: cd /usr/local
Command 9: Sudo wget https.

Command 10: Sudo tar xzf hadoop 3.3.1 tar.gz
Command 11: Sudo mv hadoop.3.3.1 hadoop
Command 12: Sudo chown -R vishakha: hadoop
Command: ls -l

Vishakha@DESKTOP-UNRAFS:/usr/local$ source ~/.bashrc
Vishakha@DESKTOP-UNRAFS:/usr/local$ cd /usr/local/hadoop/etc/hadoop
Vishakha@DESKTOP-UNRAFS:/usr/local/hadoop/etc/hadoop

Hadoop 3.3.1
Source code repository http://github.com/apache/hadoop
git -r a3bac37a397ad1880410dd80621
compiled with protoc 3.7.1
From source with checksum 88a4ddb2299aca054416d6b7
This source command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-3.1.1jar

[sudo] password for vishakha
Reading Package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional package will be installed
libwrap ncurses-term openssh-server openssh-sftp-server ssh-import-id ssh-suggested package
molly-guard monkey sphere ssh-askpass
The following New package will be installed
libwrap0 ncurses-term openssh-server
Need to get 803kB of archives
After this operation: 6290kB of additional disk space will be used.
Do you want to continue? [Y/n]Y

sudo service ssh restart
* Restarting OpenBSD Secure Shell server sshd
vishakha@DESKTOP user/local/hadoop/etc/hadoop.

DATE:

Command 13: nano $HOME/.bashrc
#Hadoop Related Options
export HADOOP_HOME=/usr/local/hadoop
export JAVA_HOME_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME

Command 14: source ~/.bashrc
Command 15: cd /usr/local/hadoop/etc/hadoop

Command 16: nano hadoop-env.sh
Command 17: sudo mkdir -p/app/hadoop/tmp
Command 18: sudo chown vishakha:hadoop /app/hadoop/tmp
Command 19: nano core-site.xml
<Property>
<name> hadoop tmp dir </name>
<Value> /app/hadoop/tmp </Value>
</Property>
<~~nam~~property>
<name> fs.default.name </name>
<Value> hdfs://localhost:54310 </Value>
</Property>
Command 20: nano mapred-site.sh
<configuration>
<property>
<name> mapred job tracker </name>
<Value> localhost:5434 </Value>
</Property> </configuration>
Command 21: nano hdfs-site.xml

WARNING: Permanently added 'localhost' (ED25519) ' to the list of known hosts. welcome to ubuntu 22.04.2 LTS
• Documentation https://help.ubuntu.com
• Management : https://landscape.canonical.com
• Support : http://ubuntu.com/advantage.

Welcome to Ubuntu 22.04.2 LTS (GNU/LINUX 5.15 901-microsoft-standard-WSL2 x86_64)
• Documentation : https://help.ubuntu.com
• Management : https://landgape.canonical.com
• Support : https://ubuntu.com/advantage
Vishakha@DESKTOP UNARAFS:

WARNING: Attempting to start all apache hadoop demons as vishakha in 10 second. WARNING: This is not a recommended production deployment configuration
WARNING: USE CTRL-C to abort
Starting namenodes on [localhost]
Starting datanodes.
Starting secondary namenodes [DESKTOP-2458RFC]
DESKTOP-2458RFC: WARNING: PARAMETER
2644 JPS
1638 NameNode
2311 NodeManager
1976 SecondaryNameNode
1787 DataNode
2187 ResourceManager

WARNING: Stopping all Apache Hadoop daemons as vishakha in seconds
WARNING: USE CTRL-C to about

DATE:

<property>
<name> dfs.~~req~~replication </name>
<property>

Command 22: cd
Command 23: hadoop version

HDFS
Command 24: cd /usr/local/hadoop/etc/hadoop
Command 25: hadoop namenode -format
Command 26: ssh
Command 27: sudo apt-get install ssh

Command 28: sudo service ssh restart

Command 29: ssh localhost

Command 30: cd/user/local/hadoop/sbin
Command 31: start-all.sh
Command 32: jps

Command 33: stop-all-sh
Thus we have successfully, installed Hadoop.

o/p :-

```
import sys
for line in sys.stdin
    line = line.strip()
    words = line.split()
    for word in words
        print ('%s\t%s' % (word, 1))
```

o/p :-

```
bus 1
Car 1
train 1
Ship 1
Ship 1
train 1
bus 1
Ship 1
Car 1
```

DATE:

Practical 2.

Aim: Implement word count program using Map Reduce.

Command : cd
Command 1: Mkdir MapReduce
Command 2: cd MapReduce
Command 3: touch word_count_data.txt
Command 4: nano word count_data.txt

```
bus car train
ship ship train
bus ship car
```

Command 5: cat word_count_data.txt
Command 6: touch mapper.py
Command 7: nano mapper.py

```
import sys
    for line in sys.stdin:
        line = line.strip()
        words = line.split()
        for word in words:
            print ('%\t%s' % (word, 1))
```

Command 8: cat word_count_data.txt Ipython: mapper.py

o/p :

```
Ship 1
bus 2
car 2
ship 2
train 2
```

Command 9: touch reducer.py
Command 10: nano reducer.py

```
from operator import itemgetter
import sys
current_word = None
current_count = 0
word = None
for line in sys.stdin
    line = line.strip()
    word, count = line.split('\t', 1)
    try
        count = int(count)
    except ValueError:
        continue
    if current_word == word:
        current_word += count
    else:
        if current_word
        print('%s\t%s' % (current_word,
            current_count))
        current_count = count
        current_word = word
if current_word == word
    print('%s\t%s' % (current_word, current_count))
```

Command 11: cat word_count_data.txt | python3 mapper.py | sort -k1,1 | python3 reducer.py

```
vishakha@DESKTOP-UNAAAF5:/usr/local/hadoop/etc/hadoop$
 Sudo service ssh restart
 * Restarting OpenBSD Secure Shell server sshd
vishakha@DESKTOP-UNAAAF5:/usr/local/hadoop/etc/hadoop$
```

```
This key is not known by any other names
Are you sure you want to continue connecting (yes/no)
Warning: Permanently added : localhost
Welcome to ubbuntu 22.04.2 LTS
* Documentation : https://help ubuntu.com
* Management : https:// landscape.conftcal.com
* Support : https//ubuntu.com/advantage
ubuntu come with ABSOLUTEIU NO WARRNIY to the extent
 permitted by application law.
* Documentation : http://help ubuntu.com
* Management : https://landscope.conanfical.com
* Support : https// ubuntu.com/advantage.
vishakha@DESKTOP-UNARAFS $
```

```
WARNING: Attempting to start all Apache Hadoop daeman
as vishakha in 10 seconds.
WARNING: This is not a recommended production deployment
configuration.
Starting NameNone on [Localhist]
Starting datarode
Starting Secondary namenode [DESKTOP-2458RFC]
2644 JPS
1638 NameNODE
2311 Node Manager
1787 DataNode
```

DATE:

Practical 3.

Aim: Exploring Hadoop Distributed file system (HDFS)

Command : su vishakha
Command: java -version
Command : hadoop version

Command 1 : cd /usr/local/hadoop/etc/hadoop
Command 2 : hadoop namenode -format
Command 3 : ssh
Command 4 : Sudo apt-get install ssh
Command 5 : Sudo service ssh restart.

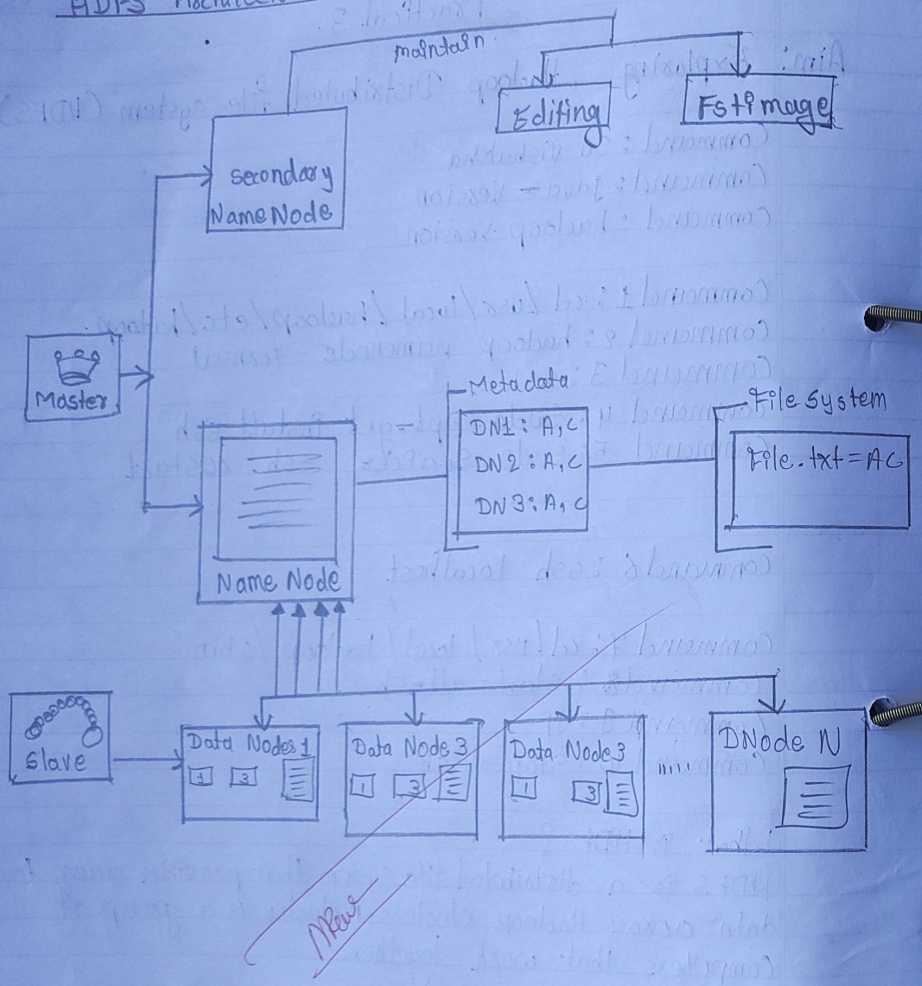
Command 6 : ssh localhost

Command 7 : cd /usr/local/hadoop/sbin
Command 8 : start-all.sh
Command 9 : jps.
Command 10 : stop-all.sh

What is HDFS?
HDFS is a distributed file system that provides access to data across Hadoop clusters. A cluster is a group of computers that work together.

Why HDFS?
Before 2011, storing and retrieving petabytes or zettabytes of data had the following three more challenges:
Cost, Speed, Reliability.

HDFS Architecture



DATE:

1. Cost: HDFS Hadoop clusters can read or write more than a terabyte of data per second.
2. Speed: Large Hadoop clusters can read or write more than a terabyte of data per second.
3. Reliability: HDFS copies the data multiple times & distributes the copies to individual nodes.

Characteristics of HDFS.

1. HDFS has high fault-tolerance.
2. HDFS may consist of thousands of server machines.
3. HDFS has high throughout.
4. HDFS is designed to store & scan millions of rows of data.
5. HDFS is designed in such a way that it can be built on commodity hardware.

HDFS Architecture and Components.

Broadly, HDFS architecture is known as the master & slave architecture which is shown below:

HDFS Components.

1. Namenode: The namenode server is the core component of an HDFS cluster. There can be only one NameNode server in an entire cluster.
2. Secondary NameNode: The secondary NameNode server maintains the edit log and namespace image information in sync with the NameNode server.
3. File system: HDFS exposes a file system namespace & allow user data to be stored in files.
4. Metadata: when new DataNodes join a cluster, metadata loads the blocks.
5. Datanode: DataNode manage names & locations of file blocks. By default, each file block is 128 Megabytes.

DATE:

Practical 4

Aim: Implement a Program in Pig.

Command 1: Su Vishakha
Command 2: java -Version
command: hadoop Version
command 2: Sudo Service ssh restart
Command 3: ssh localhost
Command 4: start-all.sh
Command 5: cd/home/~~m~~ Vishakha
Command 6: Sudo nano sample.txt

1. Sanvi, 2000, lecturer
2. Manoj, 25000, Accountant
3. Rahul, 30000, IT

Command 7: cd /usr/local
Command 8: Sudo wget https://mirrors.estointernet.in/apache/pig/pig-0.16.0/pig-0.16.0.tar.gz
Command 9: Sudo tar -xvzf pig-0.16.0.tar.gz
Command 10: Sudo mv pig-0.16.0 pig
Command 11: Sudo chmod 777 pig
command 12: ls -l

Command 13: cd /home/vishakha
Command 14: nano .bashrc

export PATH = $PATH.$HADOOP_HOME=/bin
export PATH = $PATH.$HADOOP_HOME=/sbin

DATE:

expost PIG_HOME=/usr/local/pig
expot PIG_CLASSPATH=/usr/local/hadoop/etc/hadoop

Command 15: Source .bashrc
Command 16: Pig -version

Command 17: pig
Command 18: Employees = LOAD 'Sample.txt' USING PigStorage (',') as (id: int, firstname: chararray, salary: int, dept: chararray);

Command 19: Dump Employees;
Command 20: Employees_order = ORDER Employees BY Salary DESC;

Command 21: quit

Mam

DATE:

Practical 5.

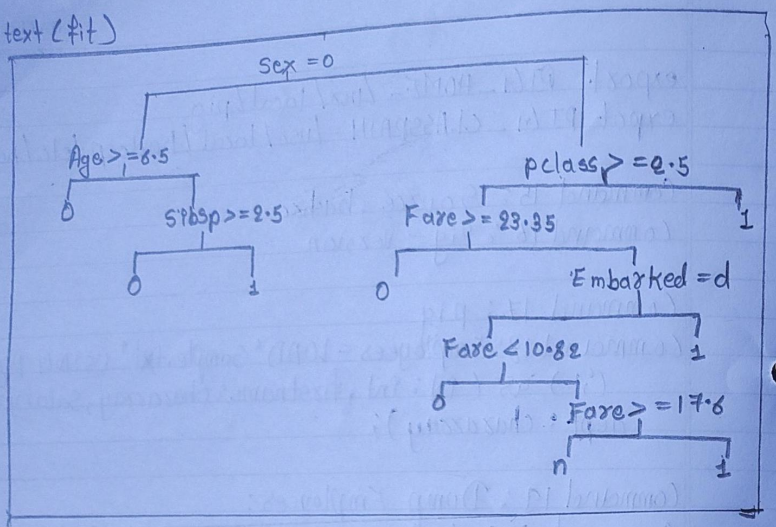
Aim : A. Implement Decision tree classification technique.
B. Implement SVM classification technique.

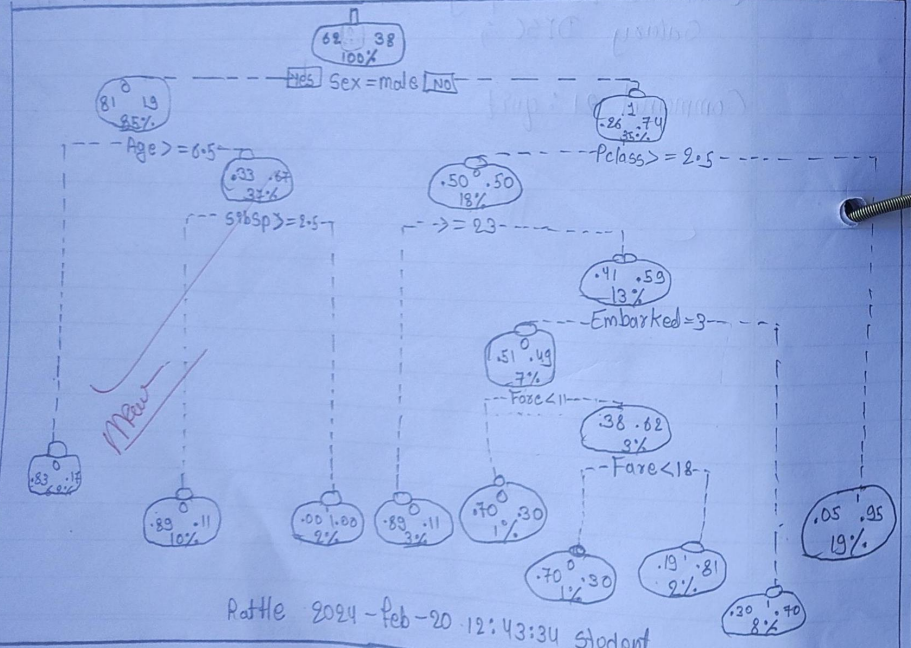
```
titanic <- read.csv(file.choose(), header=T, sep=",")
Summary(titanic)
name(titanic)
library(rpart) titanic)

library(rpart)
fit <- rpart(survived ~ Pclass + Sex + Age + sibsp + Parch +
             Fare + Embarked, data=titanic, method="class")
plot(fit)
text(fit)
install.packages('rattle')
install.packages('rpart.plot')
install.packages('RColorBrewer')
library(rattle)
library(rpart.plot)
library(RColorBrewer)
fancyRpartPlot(fit)
Prediction <- predit(fit, titanic, type="class")
Prediction
```



```
>text(fit)
```





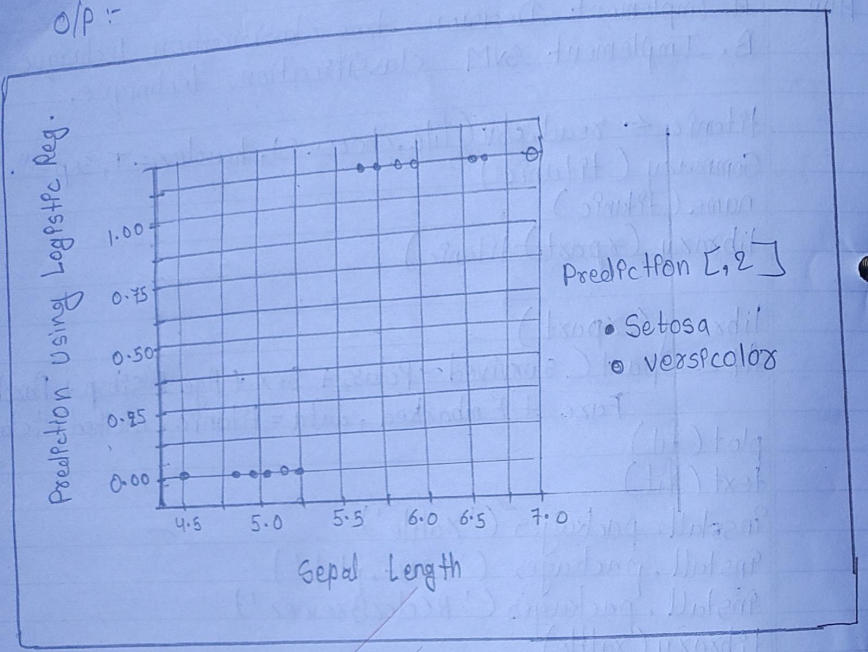
DATE:

Practical 6A

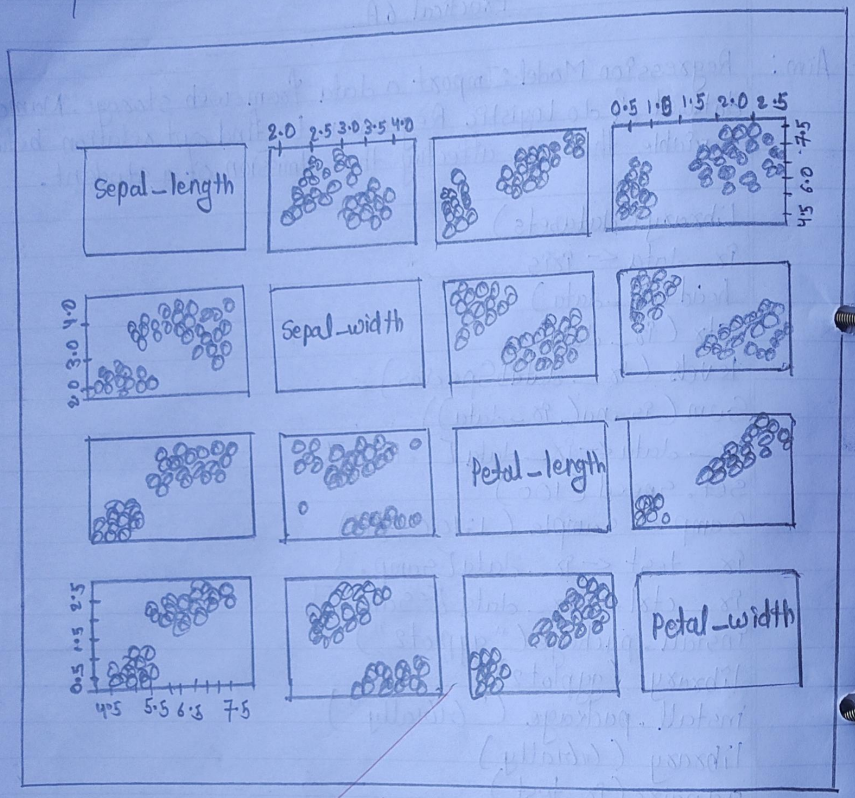
Aim: Regression Model: Import a data from web storage. Name the dataset & do Logistic Regression to find out relation between variable that are affecting the admission of a student.

```
Library (datasets)
ir_data <- iris
head (ir_data)
str (ir_data)
levels (ir_data$Species)
Sum (is.na( ir_data))
ir_data <- ir_data [1:100,]
set.speed (100)
Samp <- sample (1:100, 80)
ir_test <- ir_data[samp,]
ir_ctrl <- ir_data [-samp,]
Install.package ("ggplot2")
library (ggplot2)
Install.package ("GGally")
library (GGally)
ggpairs (ir_test)
y <- ir_test$species; x <- ir_test$Sepal.Length
glfit <- glm( y~x, family = 'binomial') summary (glfit)
newdata <- data.frame ( x = ir_ctrl$sepal.Length)
Predicted_val <- predict (glfit, newdata, type = "response")
Prediction <- data.frame (ir_ctrl$sepal.Length, ir_ctrl$Sepecies,
predicted_val)
Prediction
qplot ( prediction [,1], round (prediction [,3]), col = prediction
[,2], xlab = 'Sepal Length', ylab = 'prediction using Logistic Reg')
```

O/P :-



O/P :-



DATE:

Practical 6B

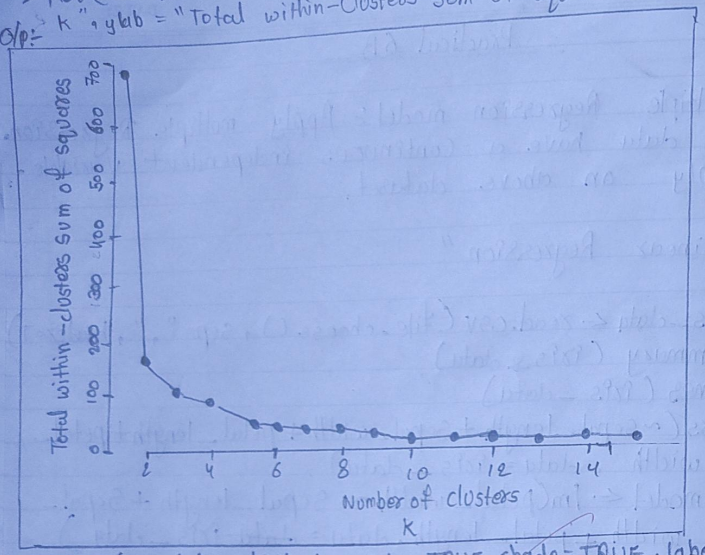
Aim : Multiple Regression model :- Apply multiple Regression, if data have a continuous independent variable. Apply an above dataset.

"Linear Regression"

```
prs_data <- read.csv(file.choose(), sep=",", header=T)
Summary (prs_data)
names (prs_data)
pairs(~Sepal_length + Sepal_width + petal_length + petal_width , data = prs_data)
prsmodel <- lm(petal_width ~ Sepal_length + Sepal_width + petal_length, data = prs_data )
Summary (prsmodel)
```

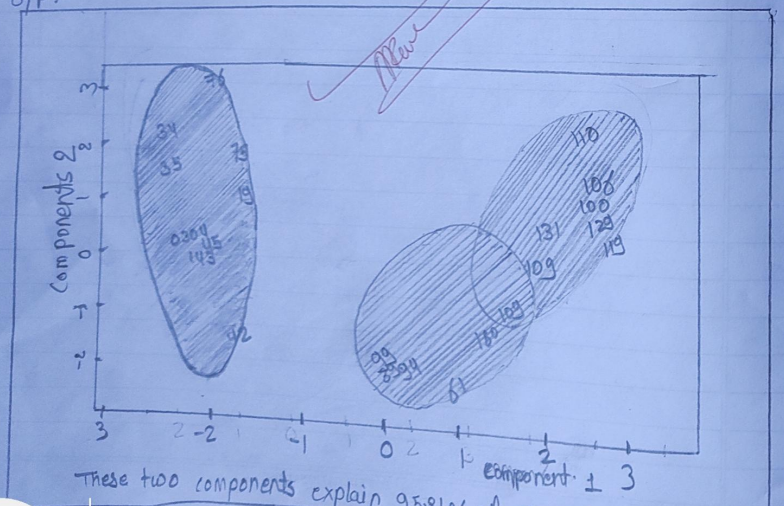
> plot(1:15, wss, type="b", pch=19, frame=FALSE, xlab="Number of cluster K", ylab="Total within-clusters sum of square")

o/p:-



> clusplot(new_data, cl$cluster, color=TRUE, shade=TRUE, labels=2, line=0)
> clu$cluster

o/p:-



These two components explain 95.81% of the point variability

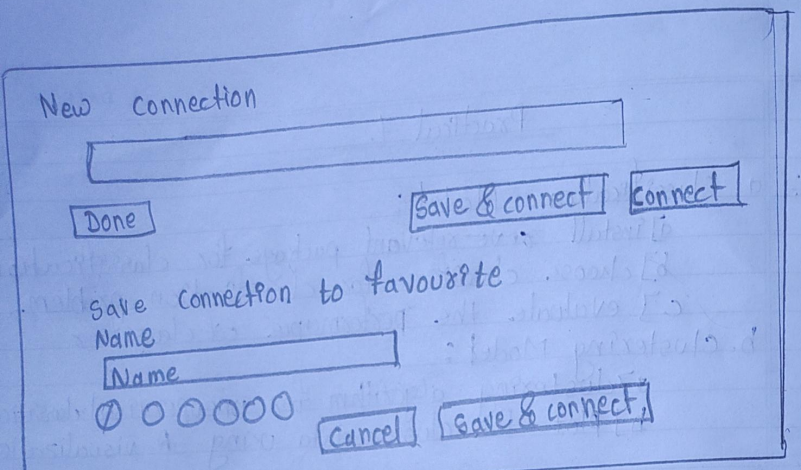
DATE:

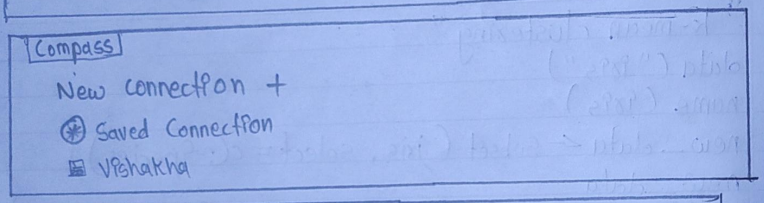
Practical 7.

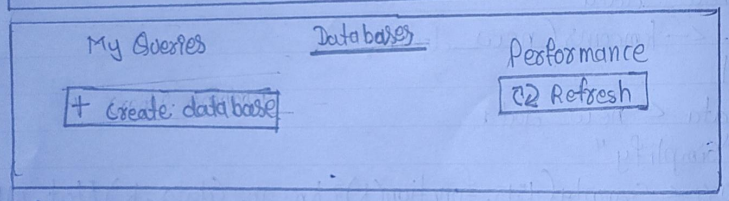
Aim:- a. Classification Model:
a] Install relevant package for classification
b] choose classifier for classification problem.
c] evaluate the performance of classifier
b. clustering Model:
a] clustering algorithm for unsupervised classification
b] Plot the cluster data using R visualizations.

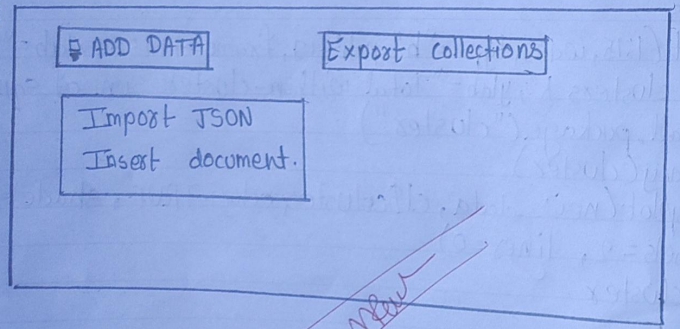
```
" k-mean clustering "
data("iris")
name(iris)
new_data <- subset(iris, select=c(-Species))
new_data
cl <- kmeans(new_data, 3)
cl
data <- new_data
"Simplify"
wss <- sapply(1:15, function(k){kmeans(data, k)$tot.withinss})
wiss
plot(1:15, wass, type="b", pch=19, frame=FALSE, xlab="Number of clusters K", ylab="Total within-clusters sum of squares")
install.package("cluster")
library(cluster)
clusplot(new_data, cl$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
cl$cluster
```

FOR EDUCATIONAL USE









DATE:

Practical 8.

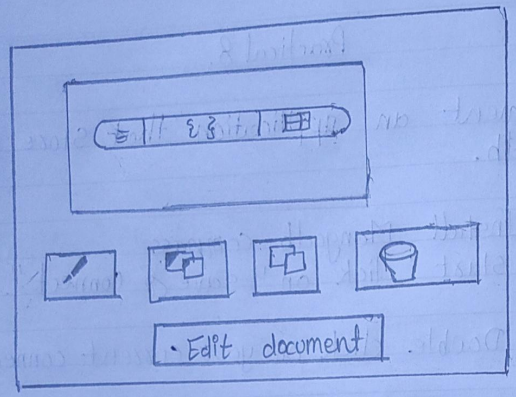
Aim:- Implement an application that store big data in Mongodb.

Step1 : Install Mongodb compass
Step 2 : start ' click on " save & connect ".

Step 3 : Double click on your current connection.

Step 4 : Move to database and ~~d~~ click on → + create database.

Step 5 : Now that the database and collection have been ~~exate~~ create . you can insert a list of unsorted document into the cities collection click an ADD DATA button.
Select the ~~Insert~~ Document option.



ADD DATA ▼ | EXPORT COLLECTION | 1 - 2 of 2

```
_id : ObjectId ("65effc 1935011 7b84ab")
name: "Mexico City"
Country: "Mexico"
Continent: "North America"
Population: 21.581
```

DATE:

Step6: Enter the following set of document into the files.

```
{"name":"Karachi","country":"Pakistan","continent:"Asia",
"population":15.400},
{"name":"Dhaka","country":"Bangladesh","continent":
"Asia","population":19.578},
{"name":"New York","country":"U.S","continent":
"North America","population":18.819},
{"name":"Manila","country":"Phillippiness",
"continent","population":14.838}
```

Step 7: Viewing, Editing and filtering the data by choosing icons.

Step8: You can also delete data & connections.