Practical Journal

**BIG DATA ANALYTICS**

**A Practical Report**

Submitted in partial fulfillment of the

Requirements for the award of the Degree

**MASTER OF SCIENCE (INFORMATION TECHNOLOGY)**

**Submitted by**

**Ms. Sokhi Rinki Jassi**

**Seat No: 1312209**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

***COSMOPOLITAN'S***

**VALIA C.L. COLLEGE OF COMMERCE &**

**VALIA L.C. COLLEGE OF ARTS**

***(Affiliated to University of Mumbai)***

**D.N. NAGAR, ANDHERI (WEST), MUMBAI 400 053**

**VALIA C.L. COLLEGE OF COMMERCE &** **VALIA L.C. COLLEGE**

**OF ARTS**

***(Affiliated to University of Mumbai)***

**D.N. NAGAR, ANDHERI (WEST), MUMBAI 400 053**

#### **DEPARTMENT OF INFORMATION TECHNOLOGY**

****

**CERTIFICATE**

This is to certify that the Journal of **"BIG DATA ANALYTICS"**, is Bonafied work of **Sokhi Rinki Jassi** bearing Seat No: **1312209** submitted in partial fulfillment of the requirements for the award of P.G of **MASTERS OF SCIENCE IN INFORMATION TECHNOLOGY** from **University of Mumbai.**

###### **Internal Guide External Examiner Co-Ordinator**

###### **Sign: Sign: Sign:**

Date: Date: Date:

Name: Name: Name:

###### 

**College Seal:**

**INDEX**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Practical No.** | **Title** | | | **Date** | **Sign** |
| **1.** | **Installation of Hadoop on Windows 10.** | | | **09/04/2024** |  |
| **2.** | **Exploring Hadoop**  **Distributed File**  **System (HDFS).** | | | **09/04/2024** |  |
| **3.** | **Store big data in Hbase / Mongodb /**  **Pig using Hadoop / R.** | | | **13/04/2024** |  |
| **4.** | **K means clustering.** | | | **16/04/2024** |  |
| **5.** | **Apriori algorithm using Groceries dataset.** | | | **16/04/2024** |  |
| **6.** | **Linear regression Model.**   1. **Simple Linear regression.** 2. **Logistic regression.** | | | **27/04/2024** |  |
| **7.** | 1. **Program to draw Decision Tree.** 2. **Naïve Bayes Classification.** | |  | **04/05/2024** |  |
| **8.** | **Text Analysis.** | |  | **11/05/2024** |  |
| **9.** | **Classification model.** | |  | **12/05/2024** |  |
| **10.** |  | **a. Implement SVM**  **Classification Technique.** |  | **12/05/2024** |  |
| **b. Clustering Model**   1. **Clustering algorithms for unsupervised classification.** 2. **Plot the cluster data using R visualizations.** |

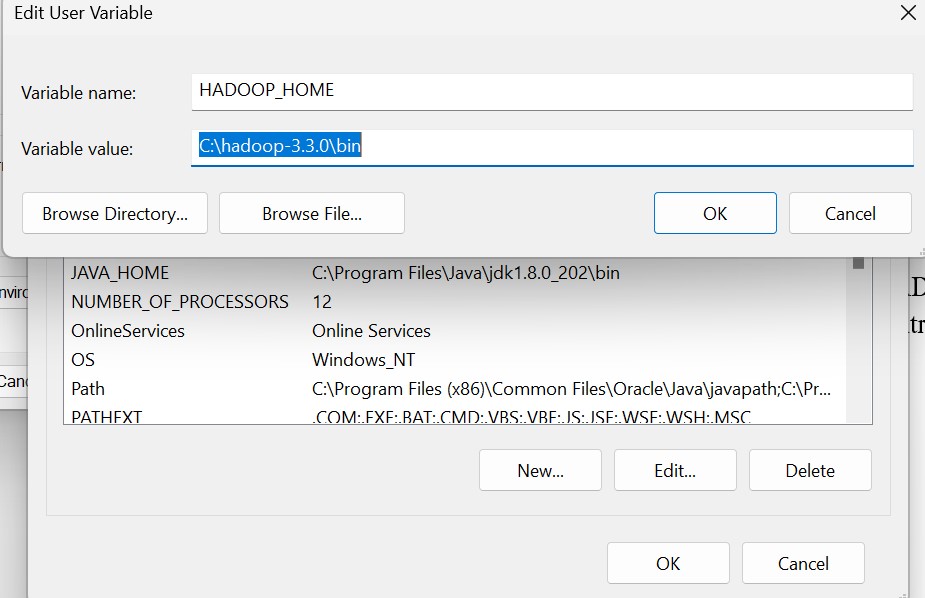
## Practical: 1

**Aim: Hadoop Installation on Windows 10.**

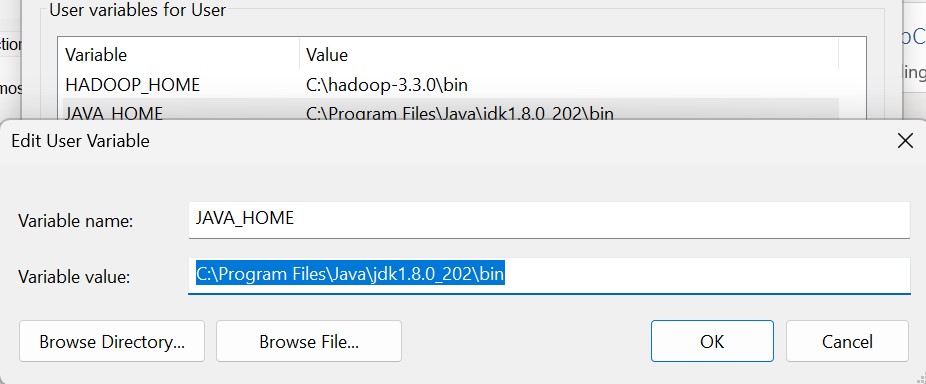
**Prerequisite**: To install Hadoop, you should have Java version 1.8 in your system.

Check your java version through this command on command prompt java -version

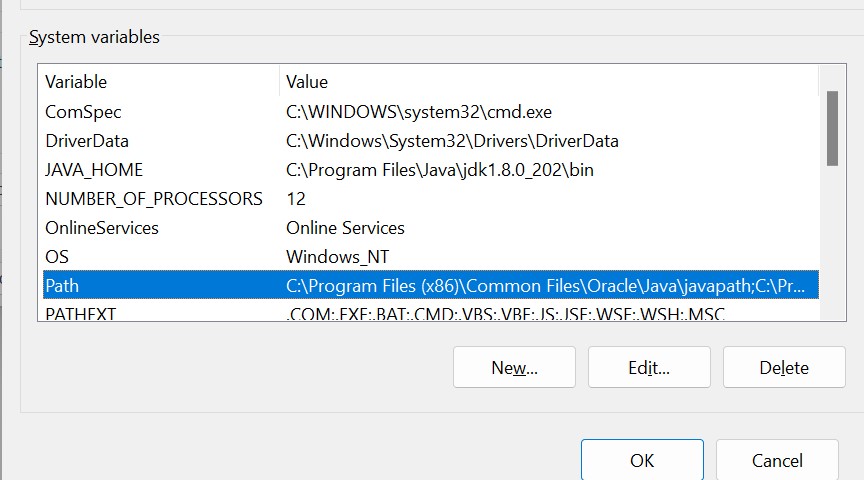
Create a new user variable. Put the Variable\_name as “HADOOP\_HOME” and Variable\_value as the path of the bin folder where you extracted hadoop.



Likewise, create a new user variable with variable name as “JAVA\_HOME” and variable value as the path of the bin folder in the Java directory.



Now we need to set **Hadoop bin** directory and **Java bin** directory path in system variable path. Edit Path in system variable.

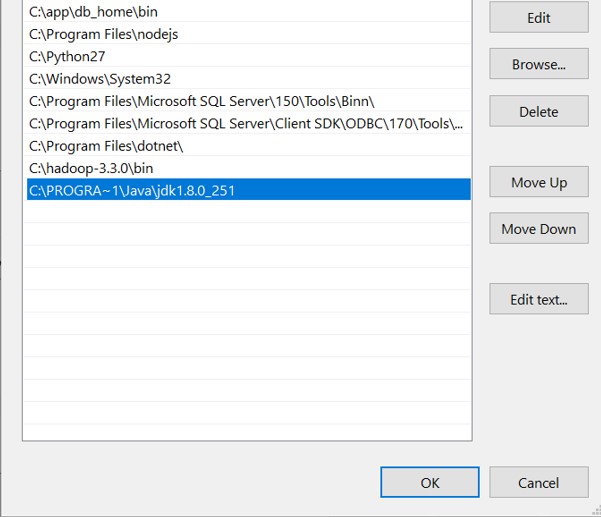


Click on New and add the bin directory path of Hadoop and Java in it.

**Hadoop**

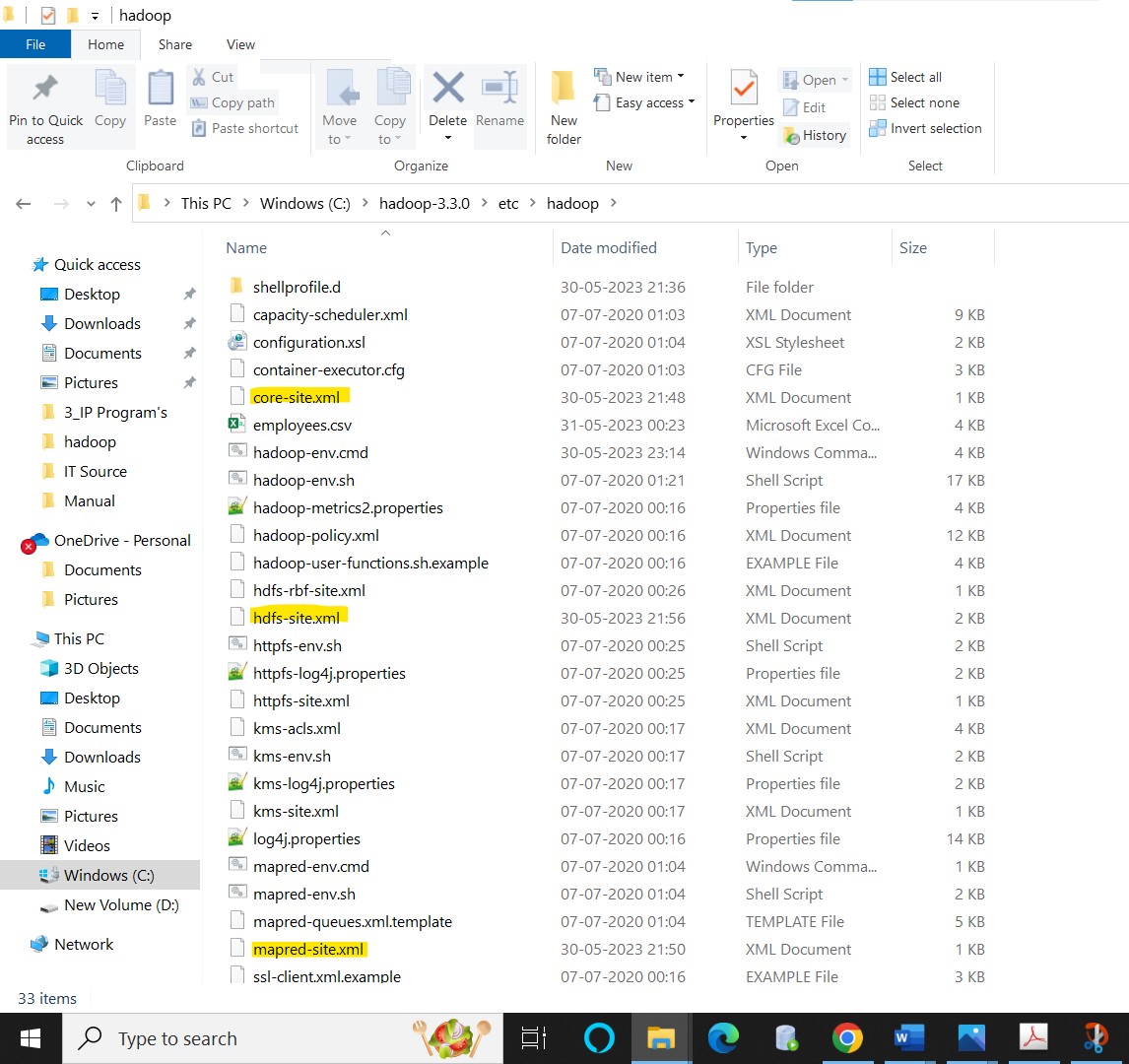


**Java**



**Configurations**

Now we need to edit some files located in the **hadoop** directory of the **etc** folder where we installed **hadoop**. The files that need to be edited have been highlighted.



1. Edit the file **core-site.xml** in the hadoop directory. Copy this xml property in the configuration in the file.

**<configuration>**

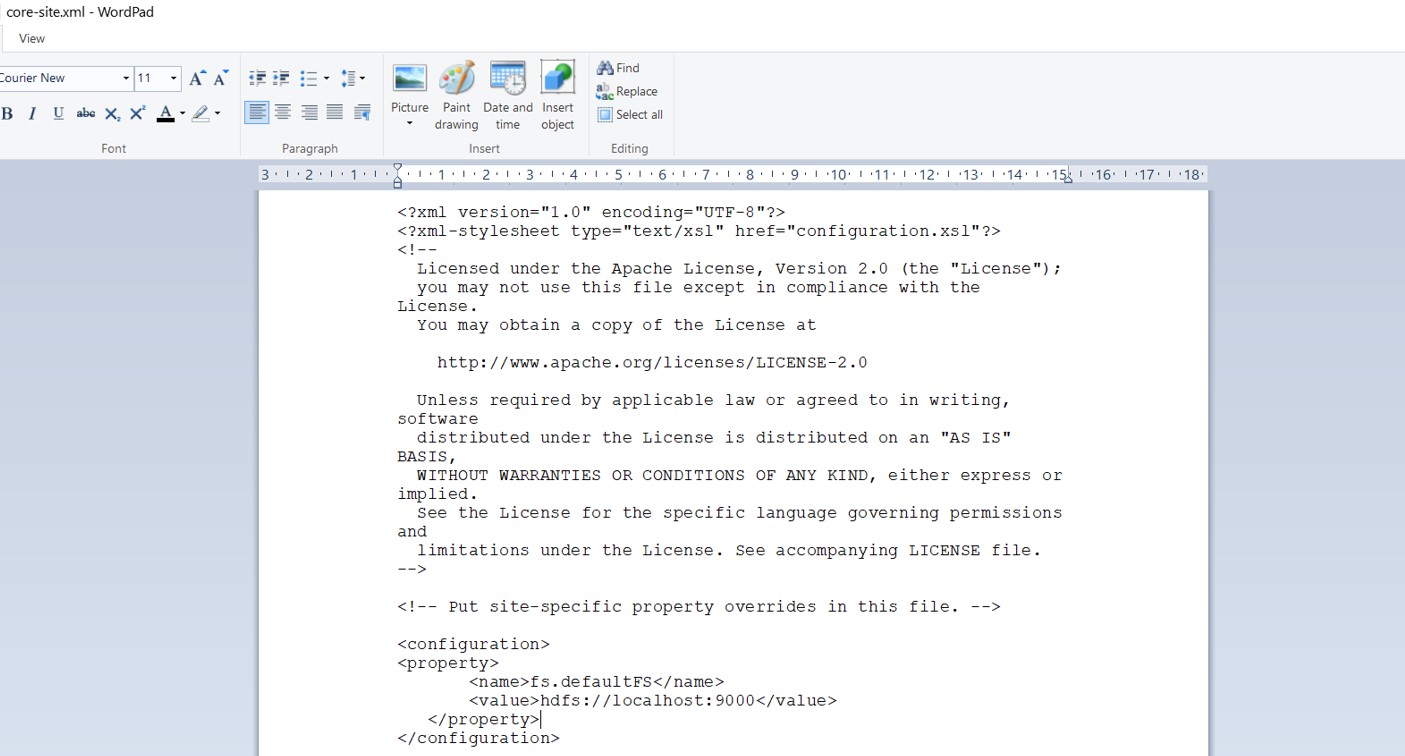
**<property>**

**<name>fs.defaultFS</name>**

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

**</property>**

**</configuration>**



1. Edit **mapred-site.xml** and copy this property in the configuration. **<configuration>**

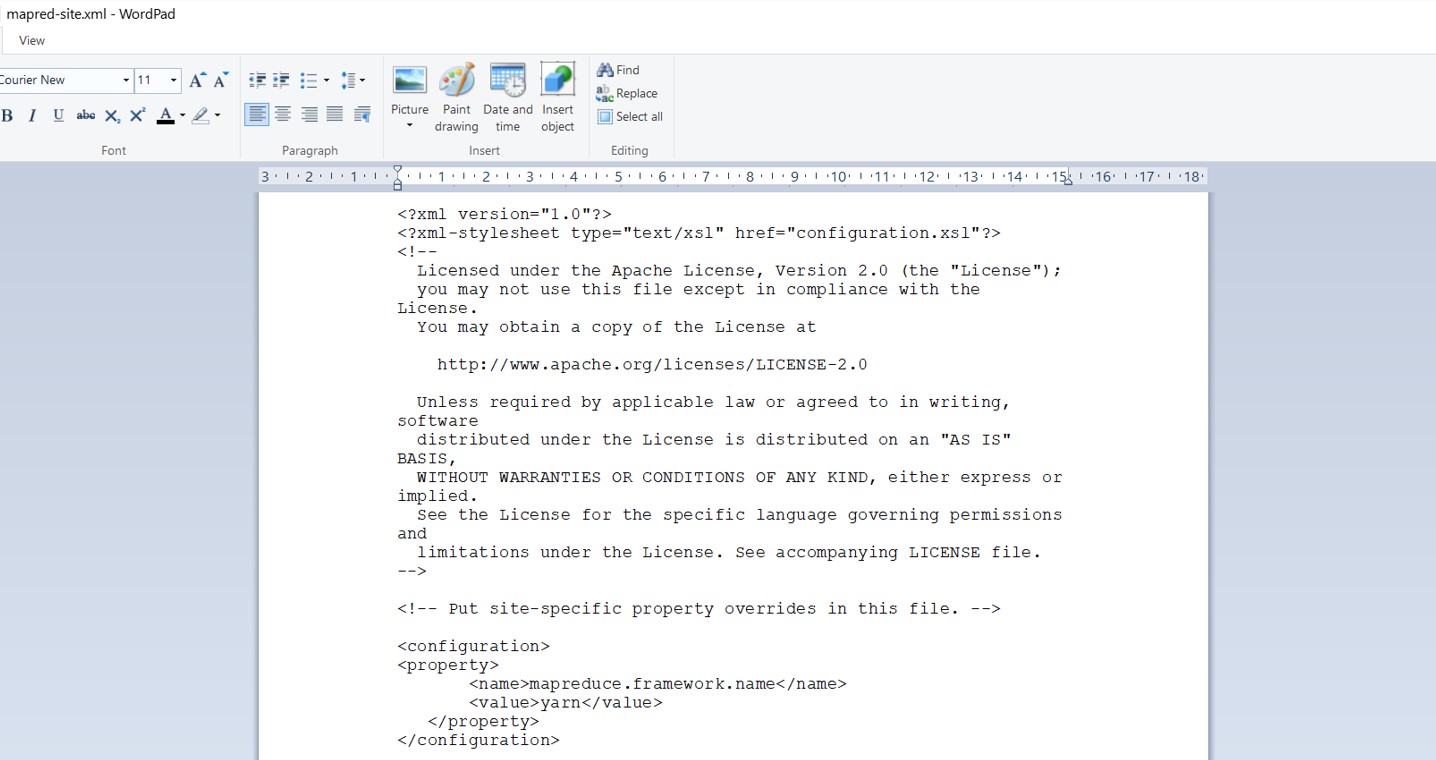
**<property>**

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

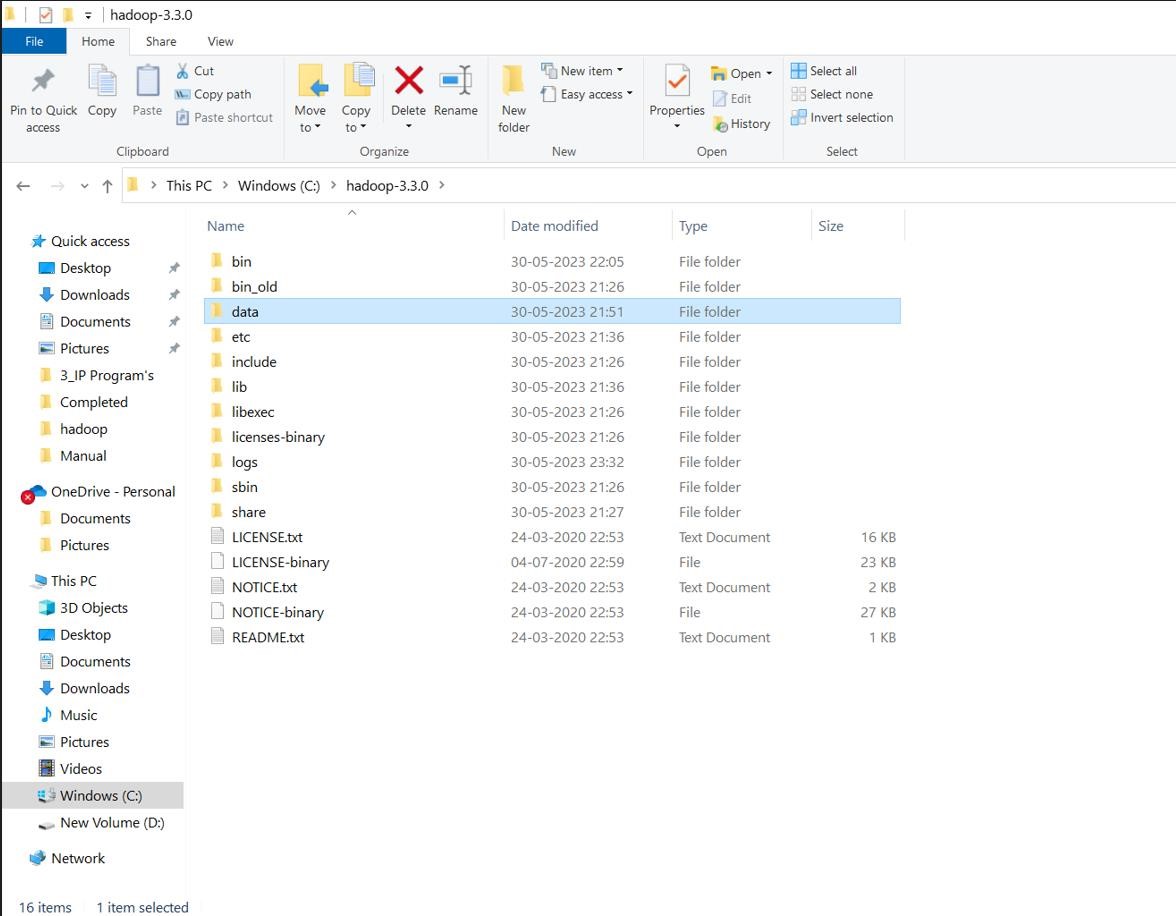
**<value>yarn</value>**

**</property>**

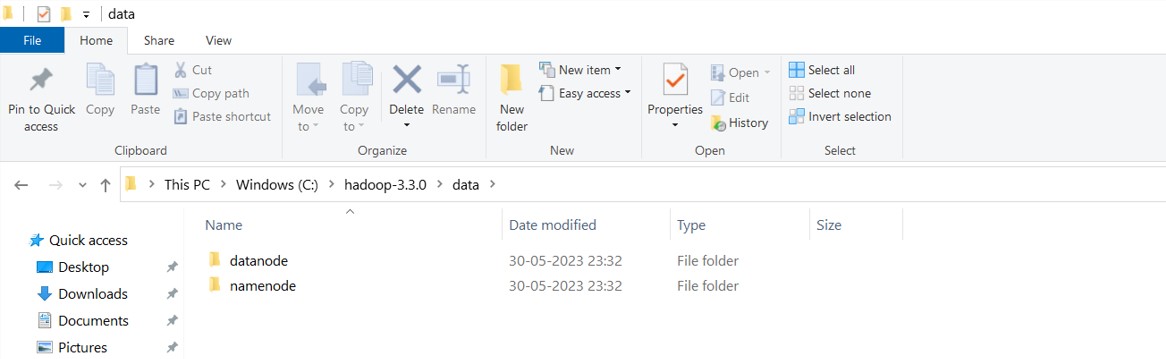
**</configuration>**



1. Create a folder **“data”** in the hadoop directory.



1. Create a folder with the name **“datanode”** and a folder **“namenode”** in this **data** directory



1. Edit the file **hdfs-site.xml** and add below property in the configuration.

**Note:** The path of namenode and datanode across value would be the path of the datanode and namenode folders you just created.

**<configuration>**

**<property>**

**<name>dfs.replication</name>**

**<value>1</value>**

**</property>**

**<property>**

**<name>dfs.namenode.name.dir</name>**

**<value>C:\hadoop-3.3.0\data\namenode</value>**

**</property>**

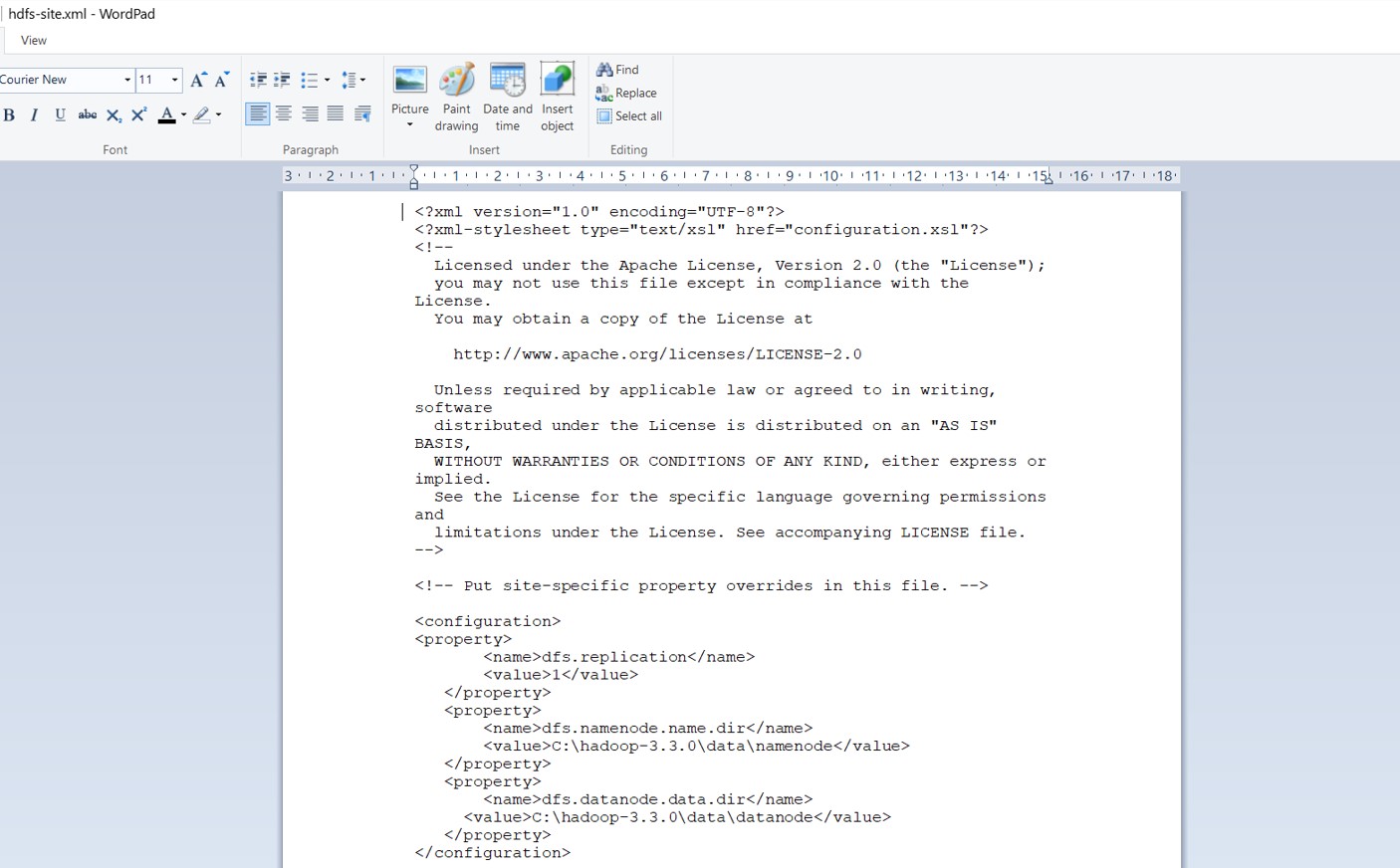
**<property>**

**<name>dfs.datanode.data.dir</name>**

**<value>C:\hadoop-3.3.0\data\datanode</value>**

**</property>**

**</configuration>**



1. Edit the file **yarn-site.xml** and add below property in the configuration. **<configuration>**

**<property>**

**<name>yarn.nodemanager.aux-services</name>**

**<value>mapreduce\_shuffle</value>**

**</property>**

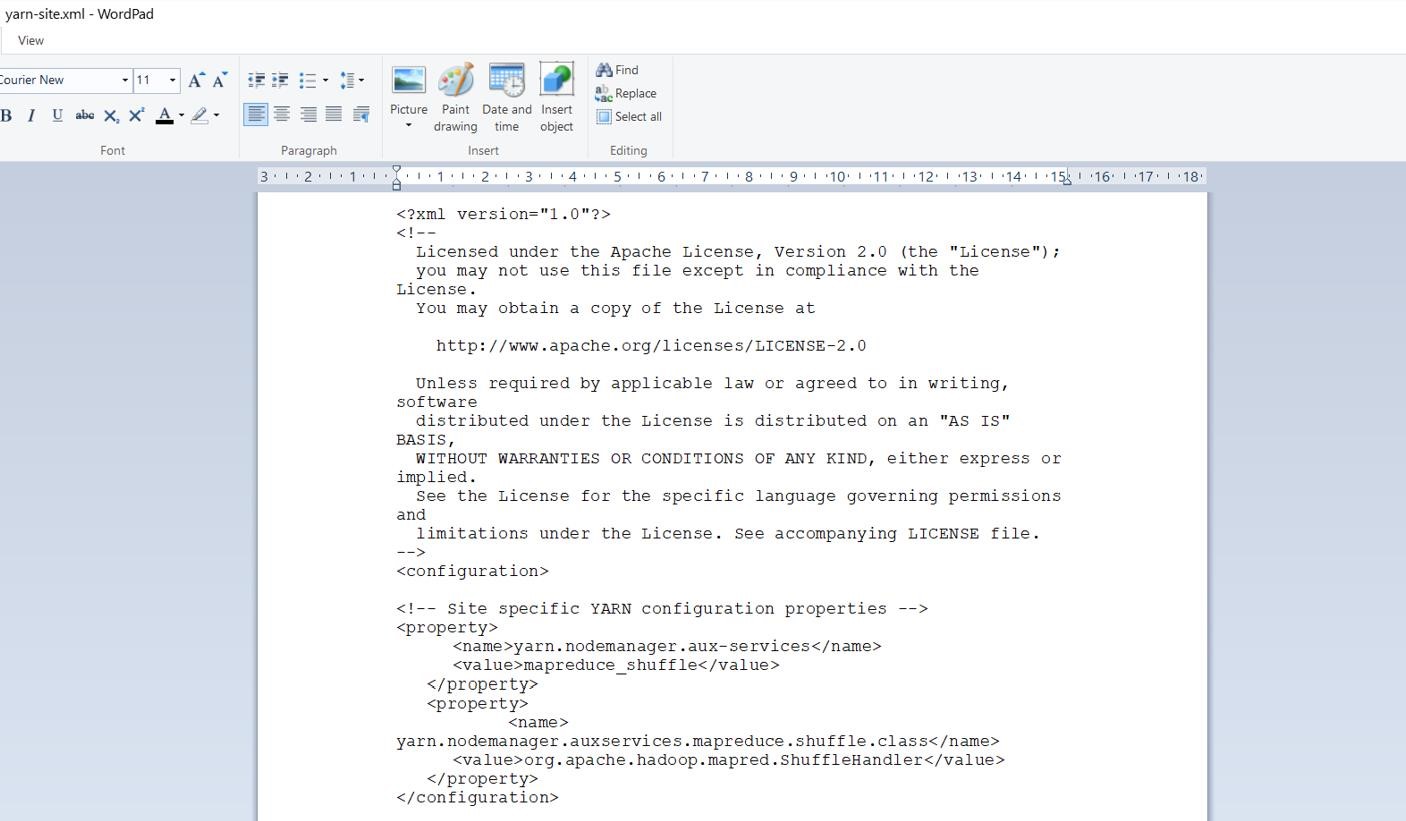
**<property>**

**<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>**

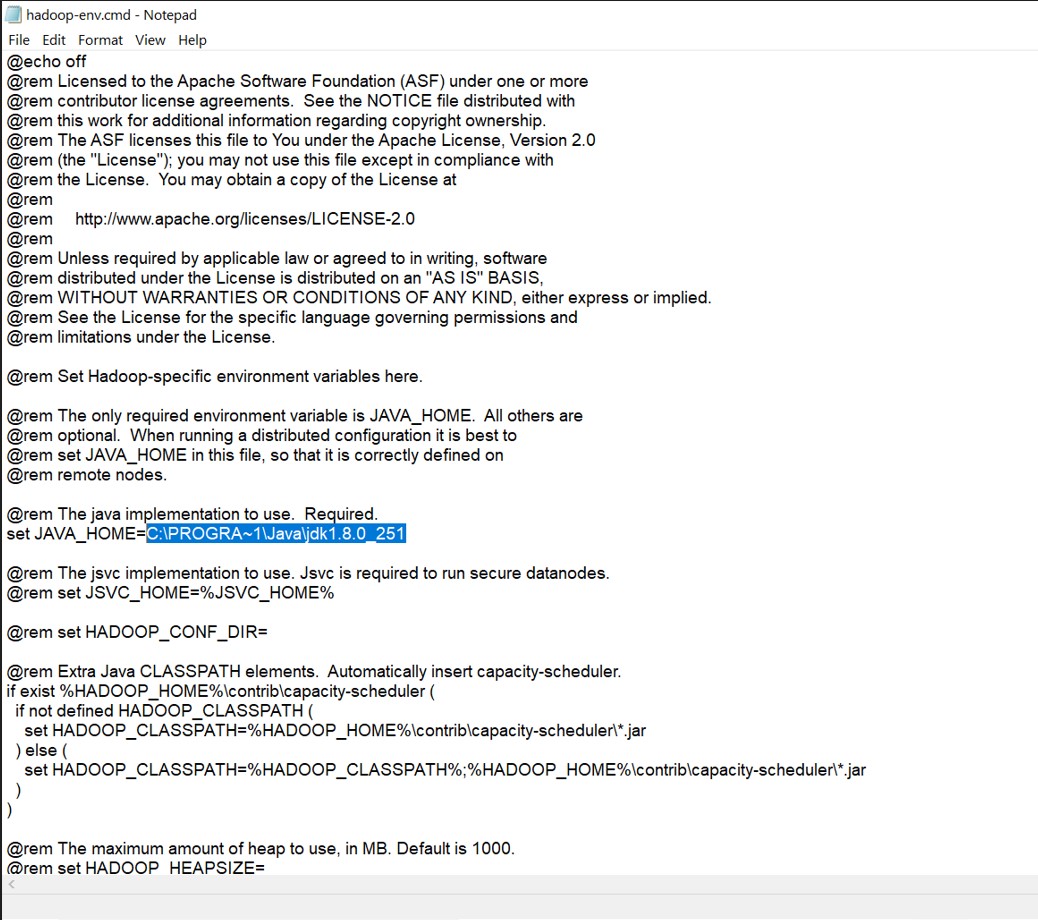
**<value>org.apache.hadoop.mapred.ShuffleHandler</value>**

**</property>**

**</configuration>**



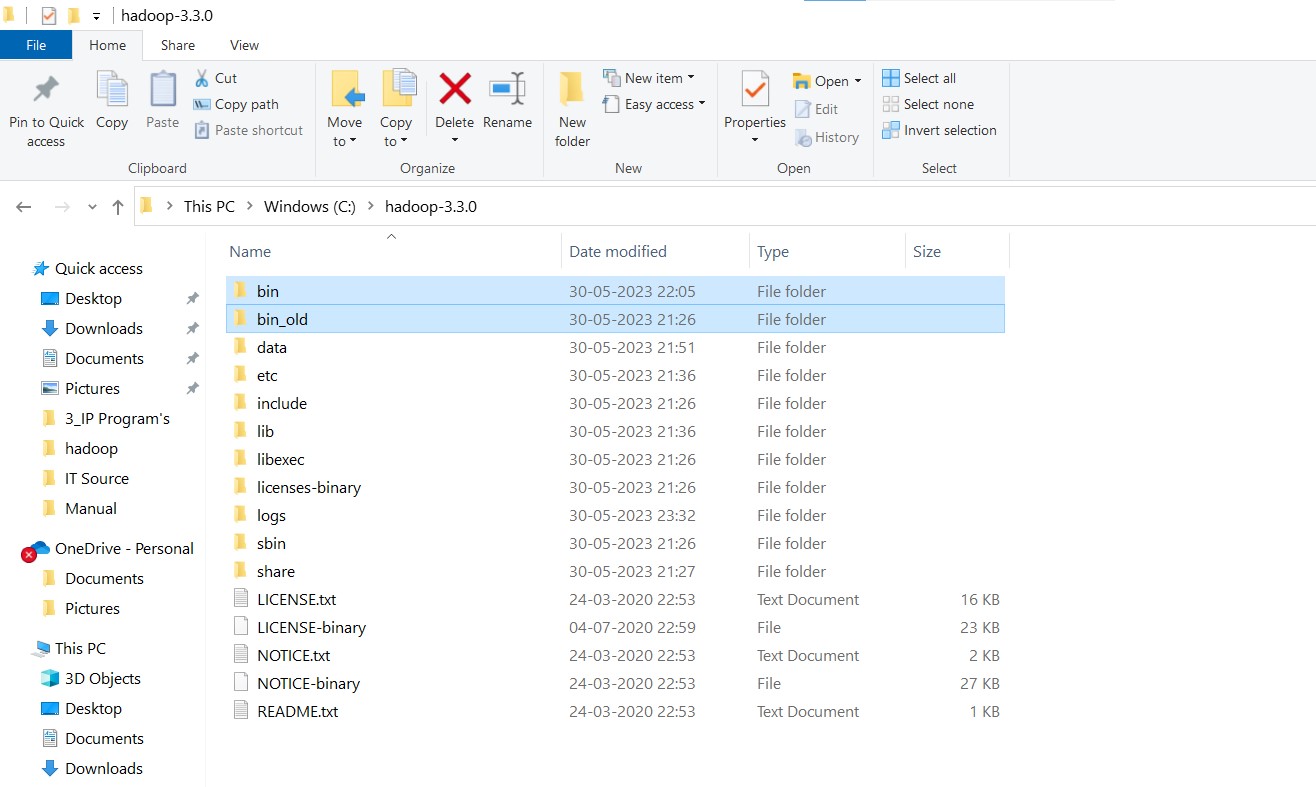
1. Edit **hadoop-env.cmd** and replace %JAVA\_HOME% with the path of the java folder where your jdk 1.8 is installed.



1. Hadoop needs windows OS specific files which does not come with default download of hadoop. To include those files, replace the bin folder in hadoop directory with the bin folder provided in this github link.

<https://github.com/s911415/apache-hadoop-3.1.0-winutils>

Download it as zip file. Extract it and copy the bin folder in it. If you want to save the old bin folder, rename it like **bin\_old** and paste the copied bin folder in that directory.



Check whether hadoop is successfully installed by running this command on cmd-**hadoop –version**

**Format the NameNode**

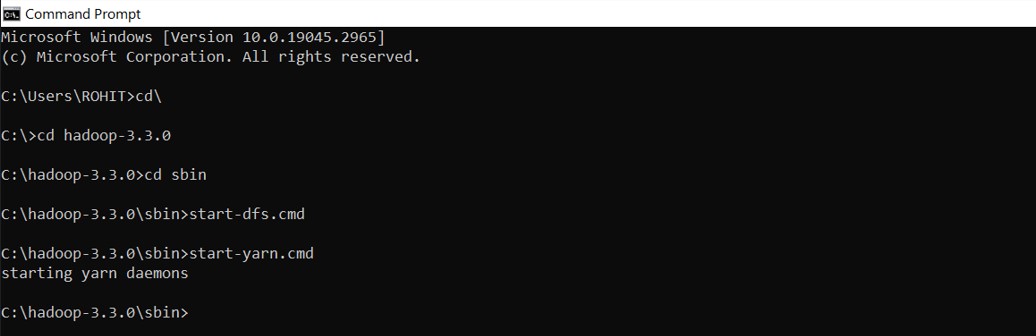
Formatting the NameNode is done once when hadoop is installed and not for running hadoop filesystem, else it will delete all the data inside HDFS. Run this command - **hdfs namenode –format**

Now change the directory in cmd to **sbin** folder of hadoop directory with this command,

Start namenode and datanode with this command – **start-dfs.cmd**

Two more cmd windows will open for **NameNode** and **DataNode**

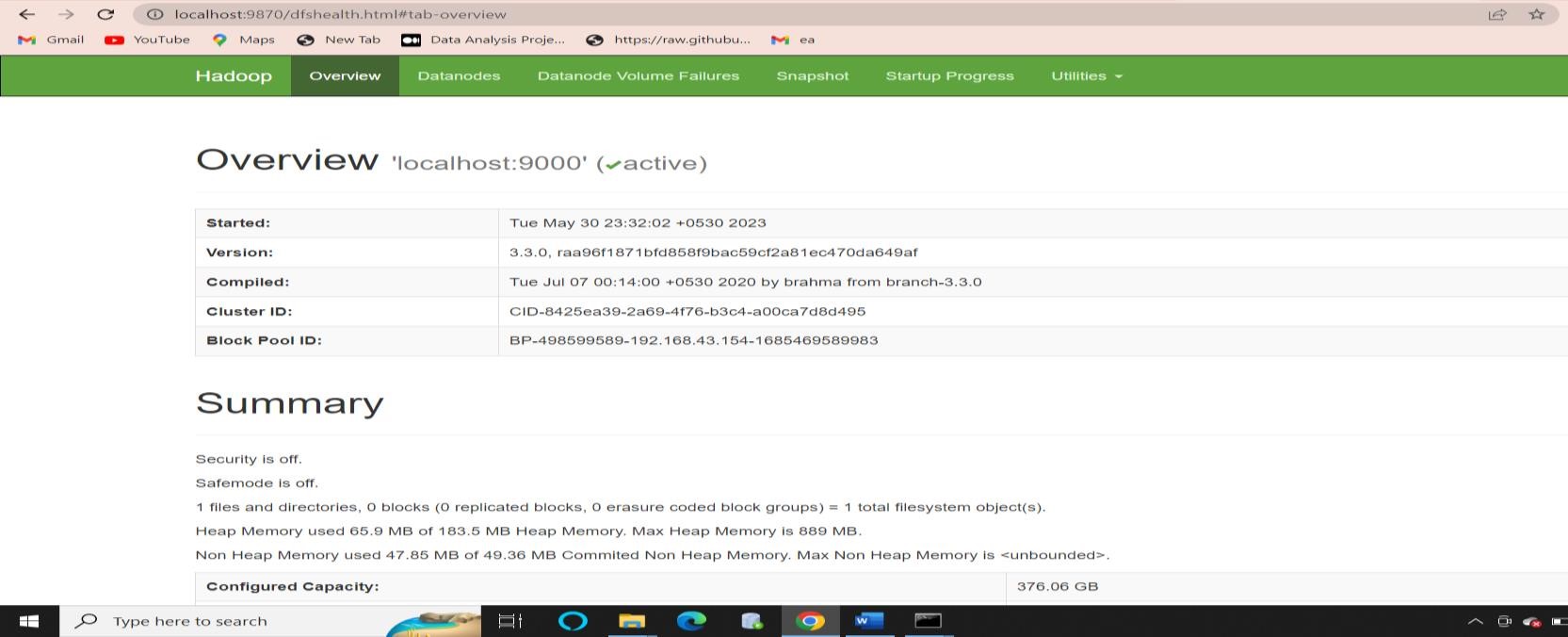
Now start yarn through this command - **start-yarn.cmd**



**Note:** Make sure all the 4 Apache Hadoop Distribution windows are up and running. If they are not running, you will see an error or a shutdown message. In that case, you need to debug the error.

To access information about resource manager current jobs, successful and failed jobs, go to this link in browser -<http://localhost:8088/cluster>

To check the details about the hdfs (namenode and datanode) -<http://localhost:9870/>



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## Practical: 2

**Aim: Exploring Hadoop Distributed File System (HDFS)**

**To implement the following file management tasks in Hadoop System (HDFS): Adding files and directories, Retrieving files, Deleting files**

**HDFS COMMANDS**

**Local File System and HDFS(Hadoop Distributed File System)**

Local file system is the file system of your own computer. Say if you are using windows, windows operating system will be having it's own way of managing files and folders. Same applies for linux.

We have also seen Hadoop has got it's own way of storing files and that is called HDFS. We have also seen, the data nodes are independent computers which gets the instruction from Name Node for storing the files. So if you think a little, a Name Node is actually using the Hadoop Distributed File System where as the Data Nodes uses it's own file system (i.e linux).

**HDFS Command line**

The two commands that helps us to interact with the HDFS are 'hadoop fs' and 'hdfs dfs'. The only difference is 'hdfs dfs' helps us to deal only with the HDFS file system and using 'hadoop fs' we can work with other file systems as well.

**1. Creating a directory in HDFS**

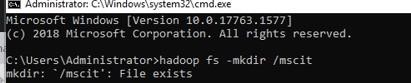
The **'mkdir'** command is used to create a directory in HDFS.

**Syntax :**

*hadoop fs -mkdir /<directory-name>*

**Example :** hadoop fs -mkdir /mscit

A directory named 'mscit' is created under the root directory.



**2. Copying files from local file system to HDFS**

To copy the files from local file system to HDFS **'copyFromLocal'** command is used.

**Syntax :** *hadoop fs -copyFromLocal <local-file-path> <hdfs-file-path>*



**Example:** hadoop fs -copyFromLocal C:\hadoop-3.3.0\etc\hadoop\employees.csv /mscit

The above command copies employee.csv from your local file system to the newly created directory

'mscit' in HDFS.

**put command**

**Syntax:** *hadoop fs -put <local-file-path> <hdfs-file-path>*

**Example:** hadoop fs -put C:\hadoop-3.3.0\etc\hadoop\employees.csv /mscit

The above command does the same thing. i.e. Copies employee.csv from your local file system to the newly created directory “mscit” in HDFS.

1. **Copying files from HDFS to local file system**

To copy the files from HDFS to local file system 'get' command is used. **Syntax:** *hadoop fs -get <hdfs-file-path> <local-file-path>*

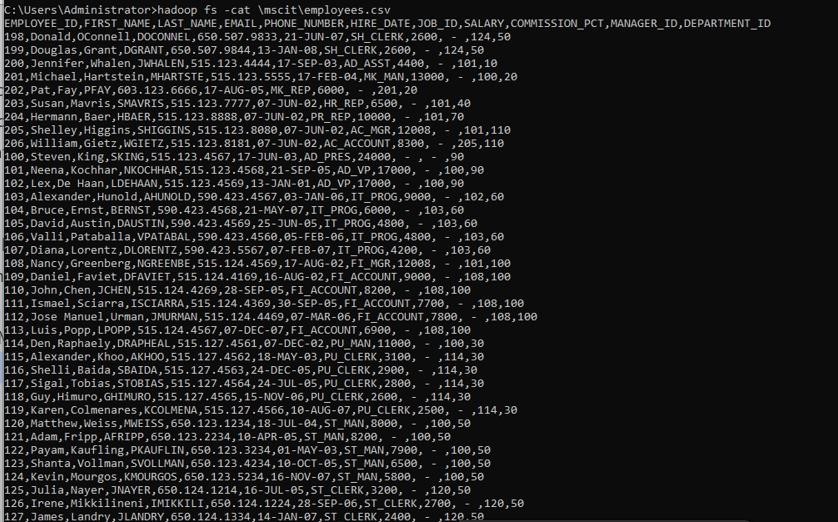
**Example:** hadoop fs -get \mscit\employee.csv C:\hadoop-3.3.0\etc\hadoop



1. **Viewing a file in HDFS**

To view a file in HDFS 'cat' command is used. **Syntax:** *hadoop fs –cat <filename>*

**Example:** hadoop fs -cat \mscit\employees.csv



1. **Display the contents of a directory in HDFS**

To display the contents if a directory in HDFS 'ls' command is used.

**Syntax:** *hadoop fs –ls <filename>*

**Example:** hadoop fs -ls /mscit

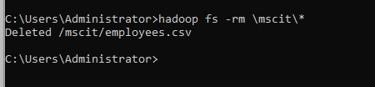


1. **Deleting all files from a directory in HDFS**

To delete files from HDFS 'rm' command is used.

**Syntax:** *hadoop fs –rm /<directory-name>/\**

**Example:** hadoop fs -rm /mscit/\*



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## Practical: 3

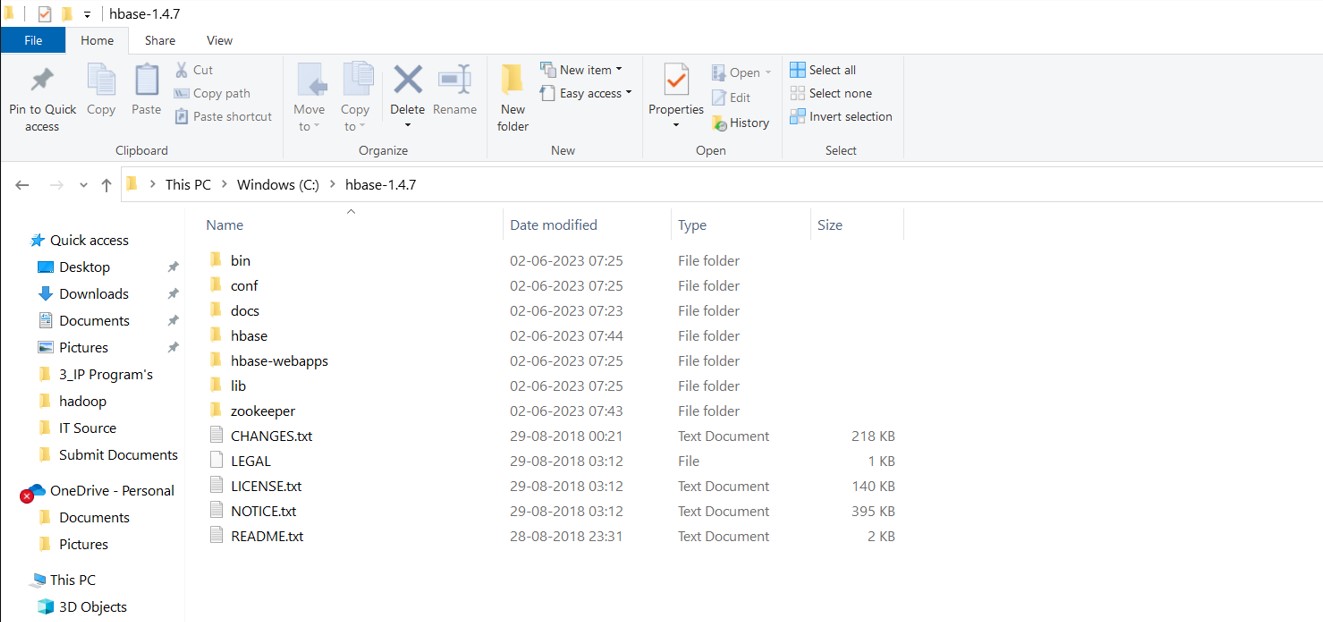
**Aim: Implement an application that store big data in Hbase/ Mongodb/ Pig using Hadoop / R.**

Hbase - Standalone mode installation.

**STEP - 1: Extract the HBase file**

**Download from:**<http://www.apache.org/dyn/closer.lua/hbase/>

Extract file hbase-1.4.7-bin.tar.gz and place under "C:\hbase-1.4.7", you can use any preferred location –

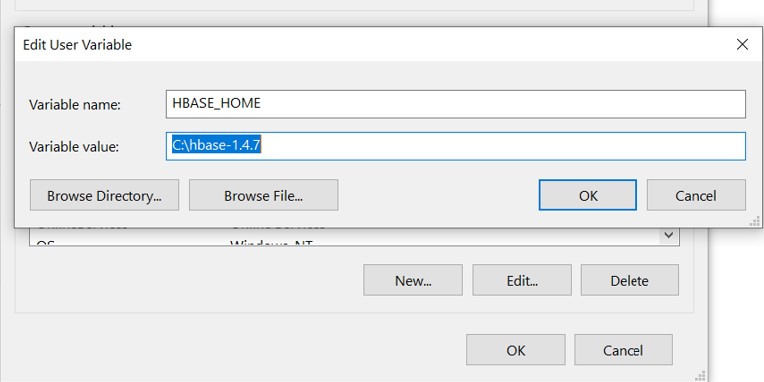


**STEP - 2: Configure Environment variable.**

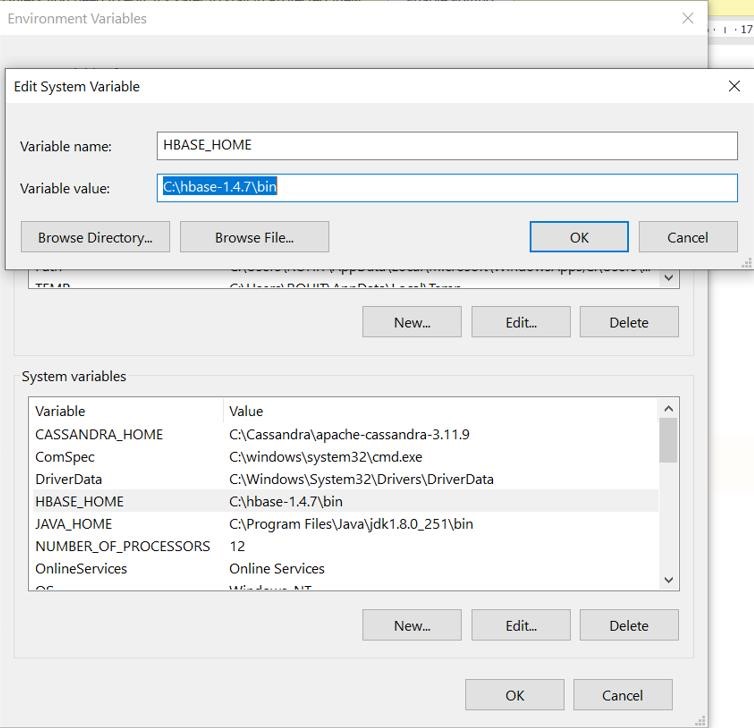
Set the path for the following Environment variable (User Variables) on windows 10 –

• HBASE\_HOME - C:\hbase-1.4.7

This PC - > Right Click - > Properties - > Advanced System Settings - > Advanced - > Environment Variables.

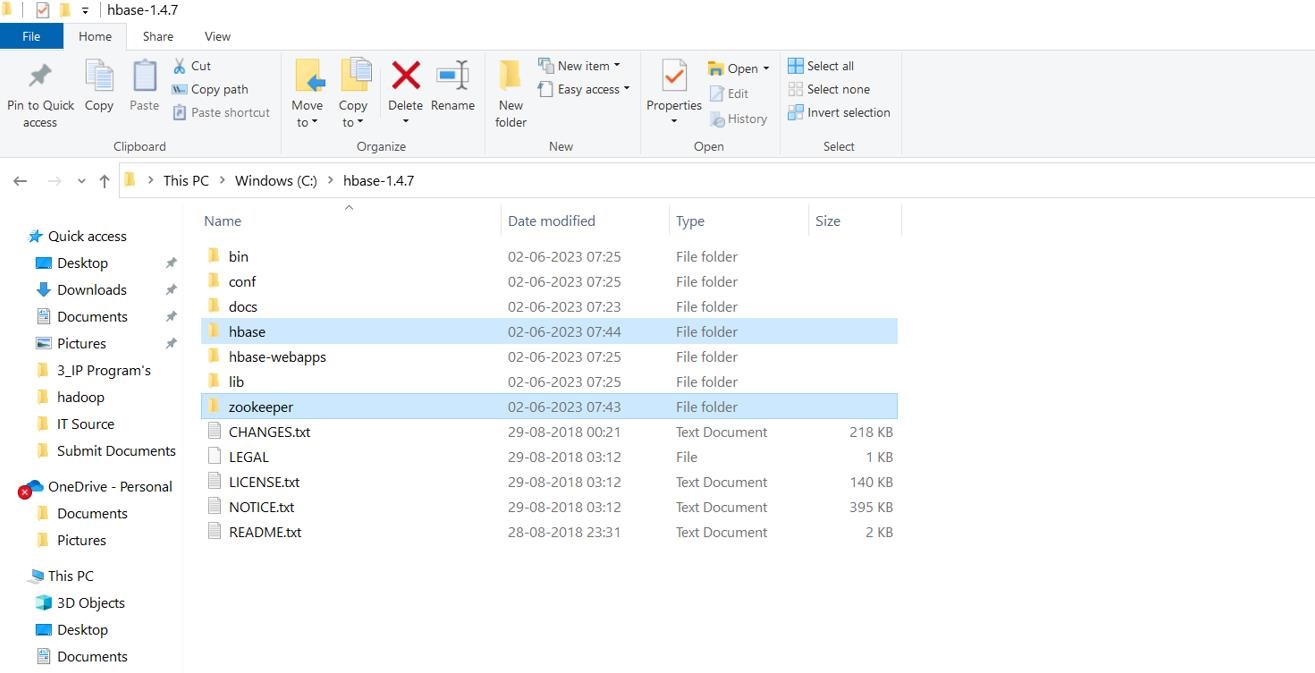


**STEP - 3: Configure System variable.**



**STEP - 4: Create required folders.**

1. Create folder "hbase" under “C:\hbase-1.4.7”.
2. Create folder "zookeeper" under “C:\hbase-1.4.7”.



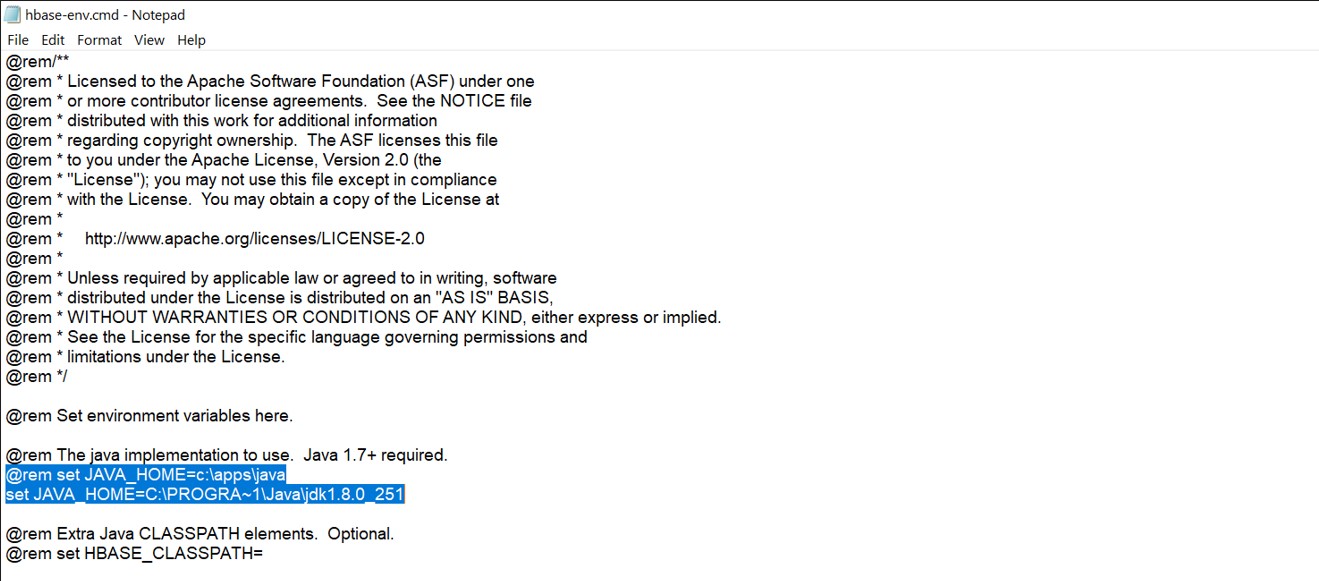
**STEP - 5: Configured required files.**

Next, essential to configure two key files with minimal required details –

• hbase-env.cmd • hbase-site.xml

1. Edit file "C:\hbase-1.4.7\conf\hbase-env.cmd", mention JAVA\_HOME path in the location and save this file.

@rem set JAVA\_HOME=c:\apps\java set JAVA\_HOME=C:\PROGRA~1\Java\jdk1.8.0\_251



1. Edit file "C:\hbase-1.4.7\conf\hbase-site.xml", paste below xml paragraph and save this file.

**<configuration>**

**<property>**

**<name>hbase.rootdir</name>**

**<value>C:\hbase-1.4.7\hbase</value>**

**</property> <property>**

**<name>hbase.zookeeper.property.dataDir</name>**

**<value>C:\hbase-1.4.7\zookeeper</value>**

**</property>**

**<property>**

**<name> hbase.zookeeper.quorum</name>**

**<value>127.0.0.1</value>**

**</property>**

**</configuration>**

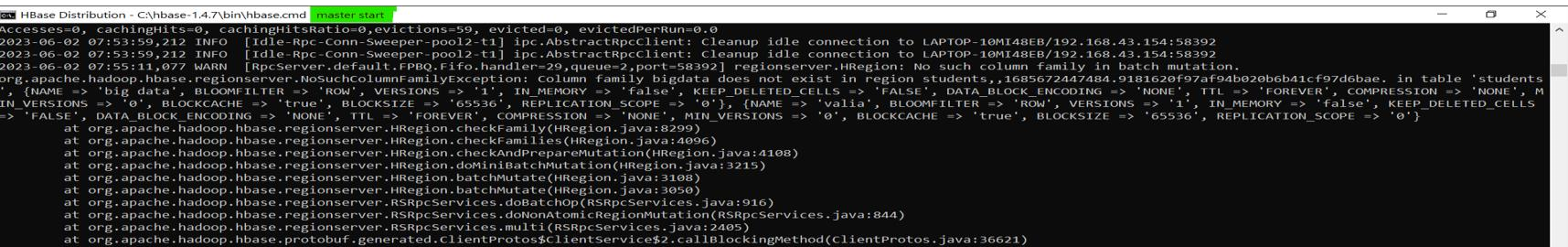
All HMaster and ZooKeeper activities point out to this hbase-site.xml.

**STEP - 6: Start HBase**

Open command prompt and change directory to “C:\hbase-1.4.7\bin" and type "start-hbase.cmd" to start HBase.



It will open a separate instances of cmd for following tasks – HBase Master



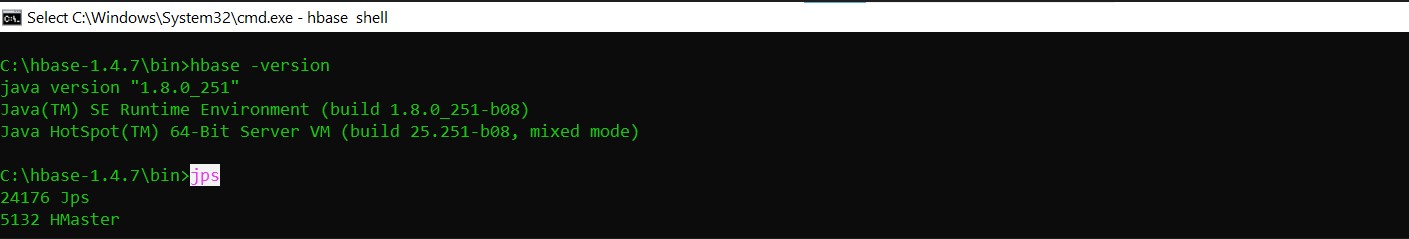
**STEP - 7: Validate HBase.**

Post successful execution of HBase, verify the installation using following commands.

* hbase –version



* jps



If we can see HMaster is in running mode, then our installation is okay.

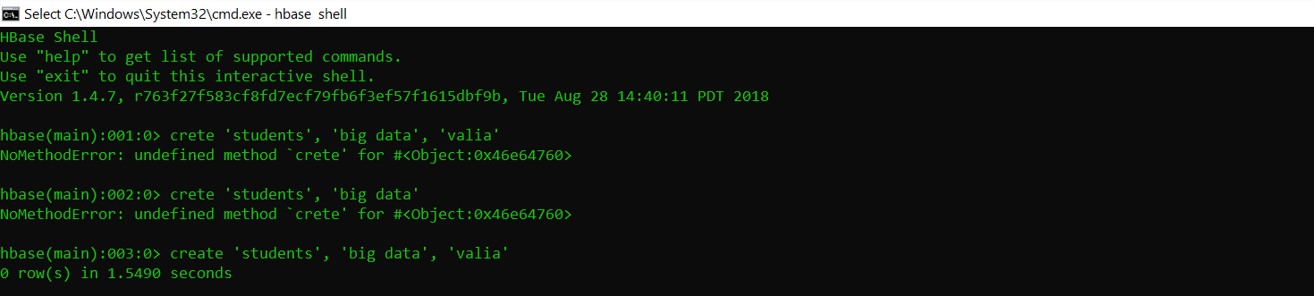
**STEP - 8: Execute HBase Shell.**

The standalone mode does not require Hadoop daemons to start. HBase can run independently. HBase shell can start by using "hbase shell" and it will enter into interactive shell mode –



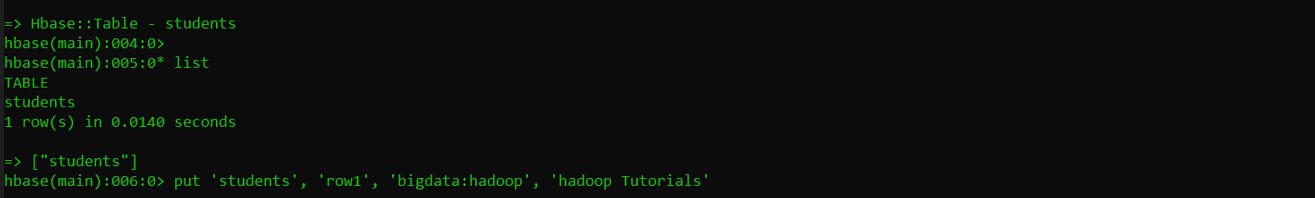
**STEP-9: Some hands on activities.**

1. **Create a simple table create 'students', 'bigdata', 'valia'**

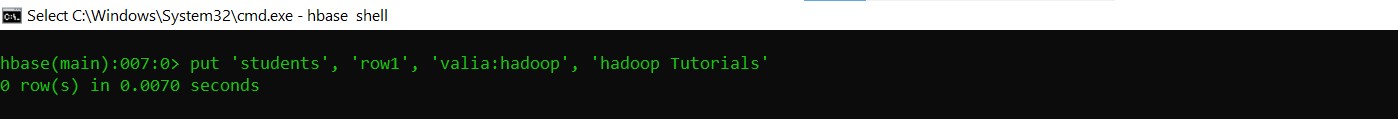


1. **List the table has been created.**

**- List**



1. **Insert some data to above created table put ‘tablename’, ‘rowname’, ‘columnvalue’, ‘value’ put 'students', 'row1', 'bigdata:hadoop', 'hadoop tutorial'**



1. **List all rows in the table scan 'students'**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

.

**Practical No: 4**

**#K means clustering.**

**Aim:** Read a datafile grades\_km\_input.csv and apply k-means clustering.

**First of all, we have to install the packages of R Program’s.**

**After that we can perform below programs.**

**install.packages("plyr") install.packages("ggplot2") install.packages("cluster") install.packages("lattice") install.packages("grid") install.packages("gridExtra")**

**# After Installation of Packages we go for main code.** library(plyr) library(ggplot2) library(cluster) library(lattice) library(grid) library(gridExtra)

grade\_input=as.data.frame(read.csv("C:\Users\Isha\Downloads\grades\_km\_input.csv" kmdata\_orig=as.matrix(grade\_input[, c ("Student","English","Math","Science")]) kmdata=kmdata\_orig[,2:4] kmdata[1:10,] wss=numeric(15)

for(k in 1:15)wss[k]=sum(kmeans(kmdata,centers=k,nstart=25)$withinss) plot(1:15,wss,type="b",xlab="Number of Clusters",ylab="Within sum of square") km = kmeans(kmdata,3,nstart=25) km

c( wss[3] , sum(km$withinss)) df=as.data.frame(kmdata\_orig[,2:4]) df$cluster=factor(km$cluster) centers=as.data.frame(km$centers)

g1=ggplot(data=df, aes(x=English, y=Math, color=cluster )) + geom\_point() + heme(legend.position="right") +

geom\_point(data=centers,aes(x=English,y=Math,

color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend =FALSE)

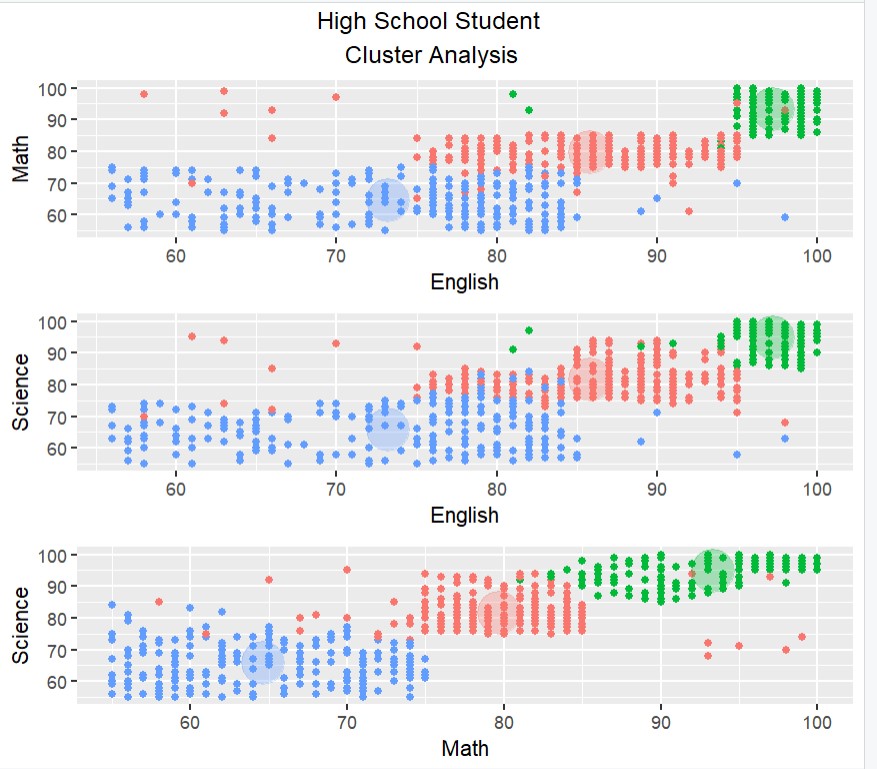
g2=ggplot(data=df, aes(x=English, y=Science, color=cluster )) + geom\_point () +geom\_point(data=centers,aes(x=English,y=Science, color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend=FALSE)

g3 = ggplot(data=df, aes(x=Math, y=Science, color=cluster )) + geom\_point () + geom\_point(data=centers,aes(x=Math,y=Science,

color=as.factor(c(1,2,3))),size=10, alpha=.3, show.legend=FALSE) tmp=ggplot\_gtable(ggplot\_build(g1))

grid.arrange(arrangeGrob(g1 + theme(legend.position="none"),g2 + theme(legend.position="none"),g3 + theme(legend.position="none"),top ="High School Student Cluster Analysis" ,ncol=1))

**Output:**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Practical No: 5**

**# Apriori algorithm.**

**Aim:** Perform Apriori algorithm using Groceries dataset from the R arules package.

**Code:**

**Packages is going to be used in this program.**

**install.packages("arules") install.packages("arulesViz") install.packages("RColorBrewer")**

library(arules) library(arulesViz) library(RColorBrewer) data(Groceries) data(Groceries) Groceries summary(Groceries) class(Groceries)

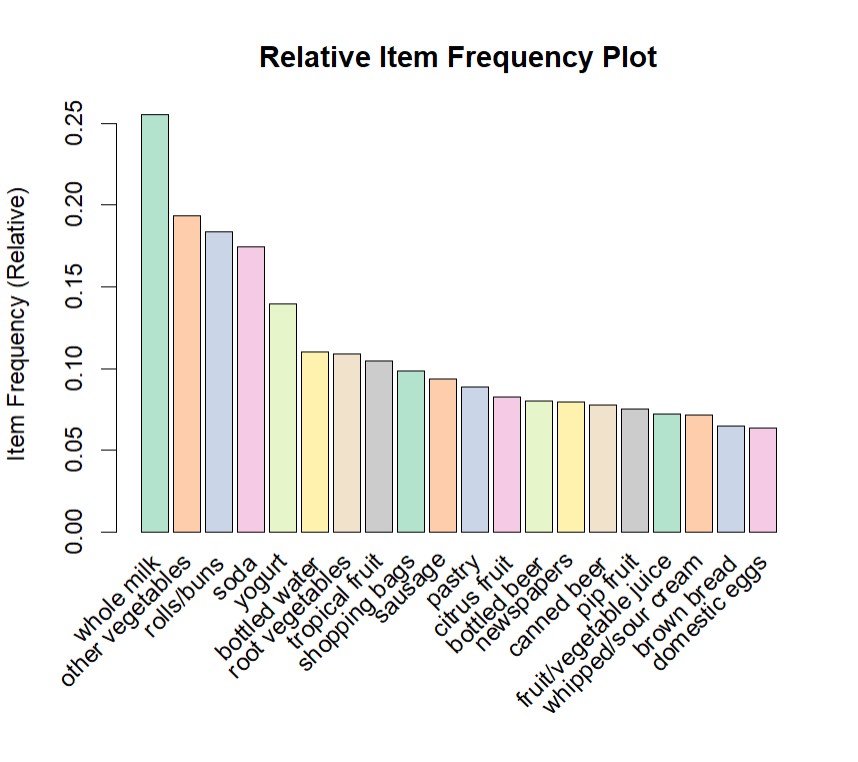
rules = apriori(Groceries, parameter = list(supp = 0.02, conf = 0.2)) summary (rules) inspect(rules[1:10])

arules::itemFrequencyPlot(Groceries, topN = 20, col = brewer.pal(8, 'Pastel2'), main = 'Relative Item Frequency Plot', type = "relative",

ylab = "Item Frequency (Relative)")

itemsets = apriori(Groceries, parameter = list(minlen=2, maxlen=2,support=0.02, target="frequent itemsets")) summary(itemsets) inspect(itemsets[1:10])

itemsets\_3 = apriori(Groceries, parameter = list(minlen=3, maxlen=3,support=0.02, target="frequent itemsets")) summary(itemsets\_3) inspect(itemsets\_3) **Output:**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Practical No: 6 # Linear Regression.**

**a. Simple Linear regression.**

**Aim:** Create your own data for years of experience and salary in lakhs and apply linear regression model to predict the salary.

**Code:**

years\_of\_exp = c(7,5,1,3) salary\_in\_lakhs = c(21,13,6,8)

#employee.data = data.frame(satisfaction\_score, years\_of\_exp, salary\_in\_lakhs)

employee.data = data.frame(years\_of\_exp, salary\_in\_lakhs) employee.data

# Estimation of the salary of an employee, based on his year of experience and satisfaction score in his company.

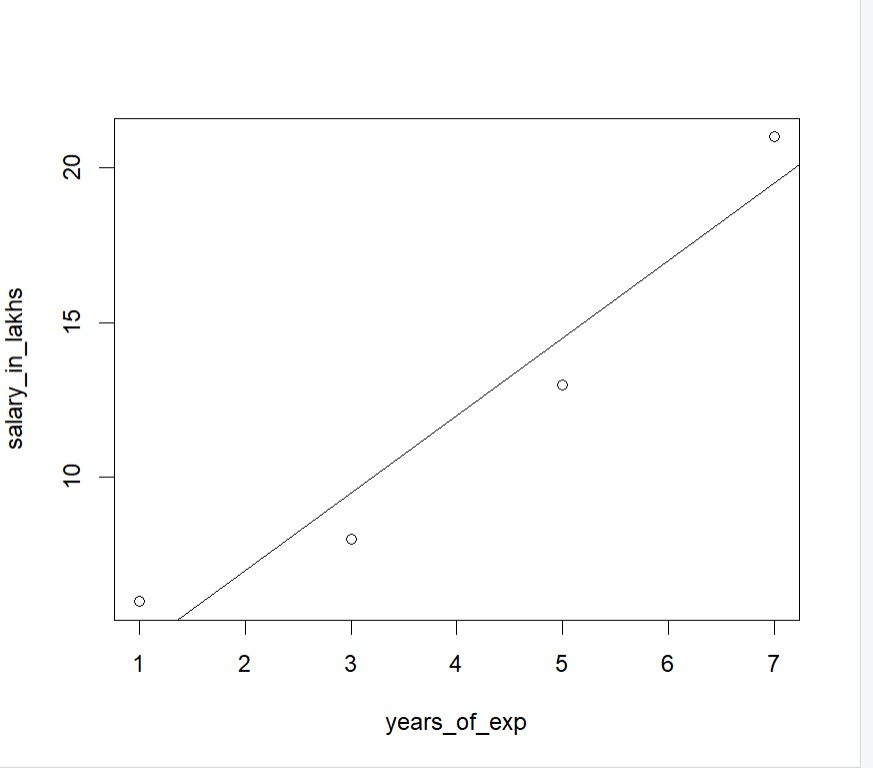
model <- lm(salary\_in\_lakhs ~ years\_of\_exp, data = employee.data) summary(model)

# The formula of Regression becomes

# Y = 2 + 2.5\*year\_of\_Exp # Visualization of Regression

plot(salary\_in\_lakhs ~ years\_of\_exp, data = employee.data) abline(model)

**Output:**



**b.** **Logistic regression.**

**Aim:** Take the in-built data from ISLR package and apply generalized logistic regression to find whether a person would be defaulter or not; considering input as student, income and balance.

**Code:**

install.packages("ISLR") library(ISLR)

data <- ISLR::Default print (head(ISLR::Default)) summary(data) nrow(data)

sample <- sample(c(TRUE, FALSE), nrow(data), replace=TRUE, prob=c(0.7,0.3)) print (sample) train <- data[sample, ] test <- data[!sample, ] nrow(train) nrow(test)

model <- glm(default~student+balance+income, family="binomial", data=train) summary(model)

#install.packages("InformationValue") library(InformationValue)

predicted <- predict(model, test, type="response") confusionMatrix(test$default, predicted)

#install.packages("confusionMatrix")

**Output:**

>

> library(ISLR)

> data <- ISLR::Default > print (head(ISLR::Default)) default student balance income 1 No No 729.5265 44361.625

1. No Yes 817.1804 12106.135
2. No No 1073.5492 31767.139
3. No No 529.2506 35704.494
4. No No 785.6559 38463.496
5. No Yes 919.5885 7491.559

> summary(data)

default student balance income No :9667 No :7056 Min. : 0.0 Min. : 772

Yes: 333 Yes:2944 1st Qu.: 481.7 1st Qu.:21340

Median : 823.6 Median :34553

Mean : 835.4 Mean :33517

3rd Qu.:1166.3 3rd Qu.:43808

Max. :2654.3 Max. :73554

> nrow(data)

[1] 10000

> sample <- sample(c(TRUE, FALSE), nrow(data), replace=TRUE, prob=c(0.7,0.

3))

> print (sample)

[1] TRUE TRUE TRUE FALSE TRUE TRUE FALSE FALSE TRUE FALSE TRUE

[12] FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

[23] FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE TRUE TRUE

[34] FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE FALSE FALSE

[45] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE

[56] TRUE TRUE TRUE TRUE FALSE TRUE FALSE FALSE TRUE FALSE TRUE

[67] FALSE FALSE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE

[78] FALSE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE TRUE TRUE

[89] TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE

[100] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE

[111] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE

[881] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE TRUE FALSE

[892] TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE TRUE

[903] TRUE FALSE TRUE TRUE FALSE FALSE TRUE FALSE TRUE TRUE TRUE

[914] FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE

[925] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

[936] TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE

[947] FALSE TRUE FALSE TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE

[958] FALSE TRUE TRUE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE

[969] TRUE TRUE FALSE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE

[980] TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE

[991] FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE TRUE

[ reached getOption("max.print") -- omitted 9000 entries ]

> train <- data[sample, ]

> test <- data[!sample, ]

> nrow(train) [1] 7010

> nrow(test)

[1] 2990

> model <- glm(default~student+balance+income, family="binomial", data=tra in)

> summary(model) Call:

glm(formula = default ~ student + balance + income, family = "binomial", data = train)

Coefficients:

Estimate Std. Error z value Pr(>|z|) (Intercept) -1.058e+01 5.769e-01 -18.340 < 2e-16 \*\*\* studentYes -7.360e-01 2.856e-01 -2.577 0.00996 \*\* balance 5.648e-03 2.738e-04 20.629 < 2e-16 \*\*\* income -1.453e-06 9.942e-06 -0.146 0.88383 ---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2003.9 on 7009 degrees of freedom

Residual deviance: 1084.4 on 7006 degrees of freedom

AIC: 1092.4

Number of Fisher Scoring iterations: 8

> #Model Diagnostics

> install.packages("InformationValue")

Warning in install.packages : package ‘InformationValue’ is not available for this version of R

A version of this package for your version of R might be available elsewhe re, see the ideas at

https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-p ackages

> library(InformationValue)

> predicted <- predict(model, test, type="response")

> confusionMatrix(test$default, predicted)

No Yes

1. 2912 64
2. 21 39

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# Practical No: 7

**# Decision Tree.**

**a. Decision Tree.**

**Code:**

dataset = read.csv "C:\\Users\\Isha\\Downloads\\Social\_Network\_Ads.csv" dataset = dataset[3:5]

# Encoding the target feature as factor dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1)) # Splitting the dataset into the Training set and Test set install.packages('caTools') library(caTools) set.seed(123) split = sample.split(dataset$Purchased, SplitRatio = 0.75) training\_set = subset(dataset, split == TRUE) test\_set = subset(dataset, split == FALSE)

# Feature Scaling training\_set[-3] = scale(training\_set[-3]) test\_set[-3] = scale(test\_set[-3])

# Fitting Decision Tree Classification to the Training set install.packages('rpart')

library(rpart) classifier = rpart(formula = Purchased ~ ., data = training\_set)

# Predicting the Test set results y\_pred = predict(classifier, newdata = test\_set[-3], type = 'class')

# Making the Confusion Matrix cm = table(test\_set[, 3], y\_pred) # Visualising the Training set results install.packages("ElemStatLearn") library(ElemStatLearn) set = training\_set

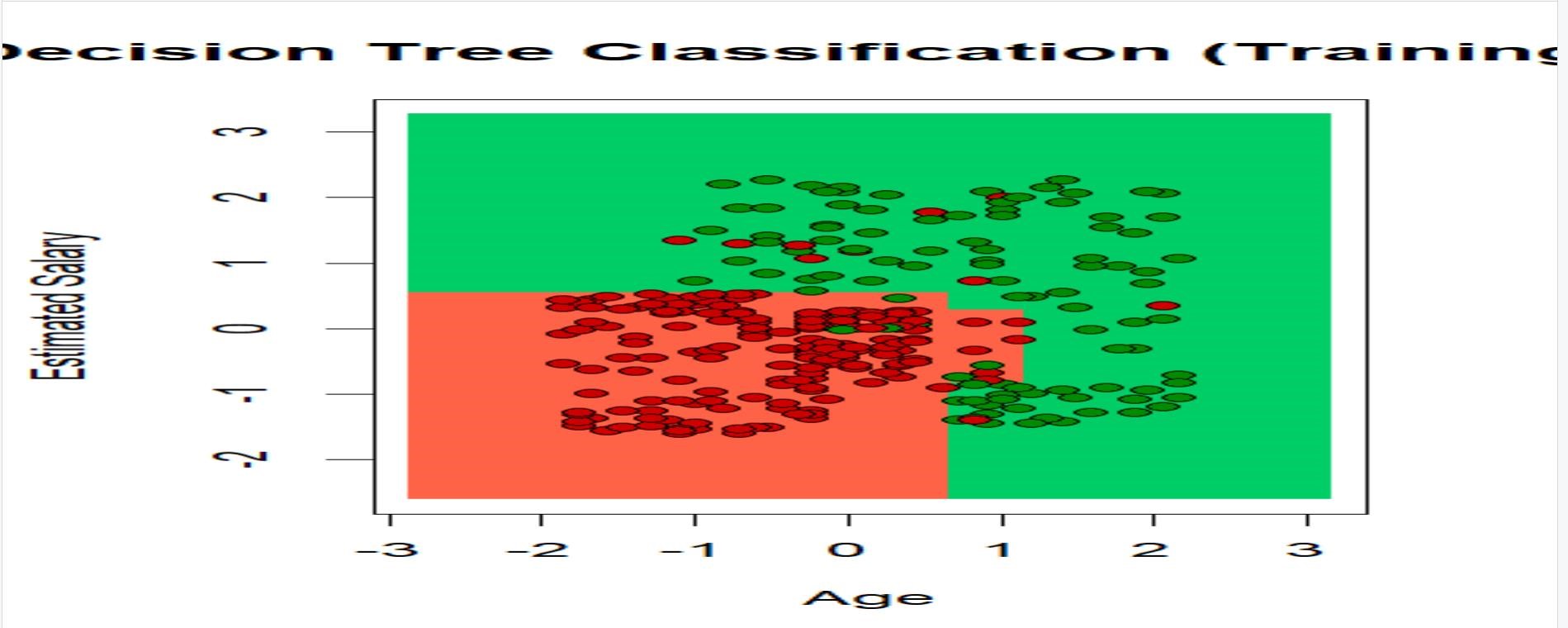
X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01) X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01) grid\_set = expand.grid(X1, X2) colnames(grid\_set) = c('Age', 'EstimatedSalary') y\_grid = predict(classifier, newdata = grid\_set, type = 'class')

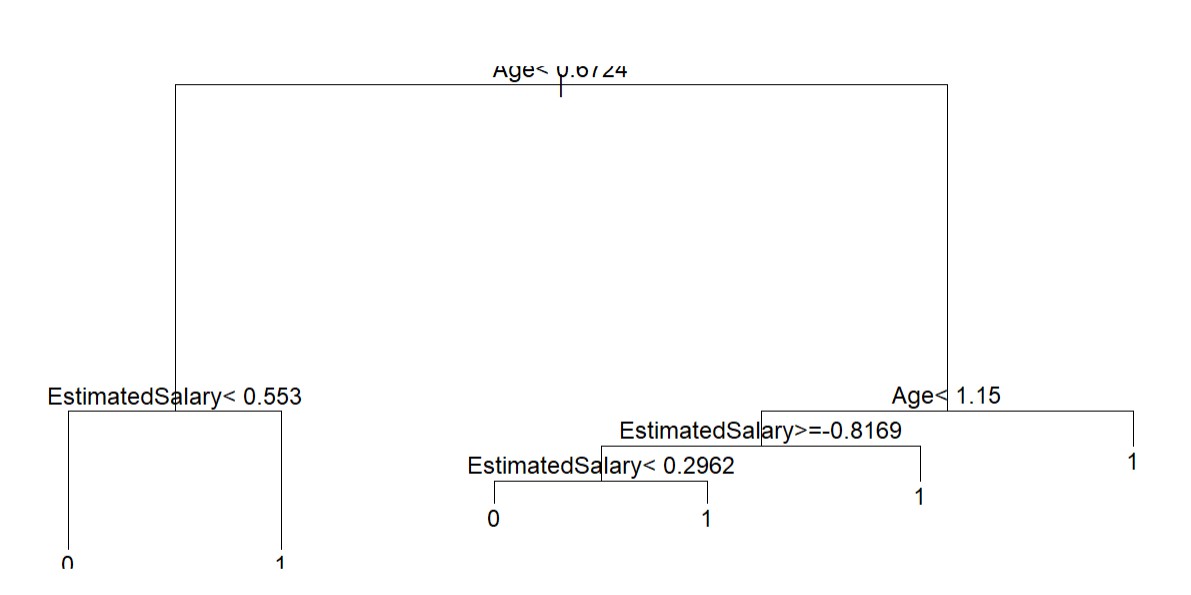
plot(set[, -3], main = 'Decision Tree Classification (Training set)', xlab = 'Age', ylab = 'Estimated Salary', xlim = range(X1), ylim = range(X2)) contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE) points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'springgreen3', 'tomato')) points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))

# Visualising the Test set results library(ElemStatLearn) set = test\_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01) X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01) grid\_set = expand.grid(X1, X2) colnames(grid\_set) = c('Age', 'EstimatedSalary') y\_grid = predict(classifier, newdata = grid\_set, type = 'class') plot(set[, -3], main = 'Decision Tree Classification (Test set)', xlab = 'Age', ylab = 'Estimated Salary', xlim = range(X1), ylim = range(X2)) contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE) points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'springgreen3', 'tomato')) points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))

# Plotting the tree plot(classifier) text(classifier) **Output:**





**# Naïve Bayes Classification.**

**b. Naïve Bayes Classification.**

**Code:**

dataset = read.csv"C:\Users\Isha\Downloads\Social\_Network\_Ads.csv") dataset = dataset[3:5]

# Encoding the target feature as factor dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1))

# Splitting the dataset into the Training set and Test set

#install.packages('caTools') library(caTools) set.seed(123) split = sample.split(dataset$Purchased, SplitRatio = 0.75) training\_set = subset(dataset, split == TRUE) test\_set = subset(dataset, split == FALSE)

# Feature Scaling training\_set[-3] = scale(training\_set[-3]) test\_set[-3] = scale(test\_set[-3]) # Fitting Naive Bayes to the Training set install.packages('e1071') library(e1071) classifier = naiveBayes(x = training\_set[-3], y = training\_set$Purchased)

# Predicting the Test set results y\_pred = predict(classifier, newdata = test\_set[-3]) # Making the Confusion Matrix cm = table(test\_set[, 3], y\_pred) print(cm)

# Visualising the Training set results install.packages("ElemStatLearn") library(ElemStatLearn) set = training\_set print(set)

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01) X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01) grid\_set = expand.grid(X1, X2) colnames(grid\_set) = c('Age', 'EstimatedSalary') y\_grid = predict(classifier, newdata = grid\_set)

plot(set[, -3], main = 'Naive Bayes (Training set)', xlab = 'Age', ylab = 'Estimated Salary', xlim = range(X1), ylim = range(X2)) contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE) points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'springgreen3', 'tomato')) points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))

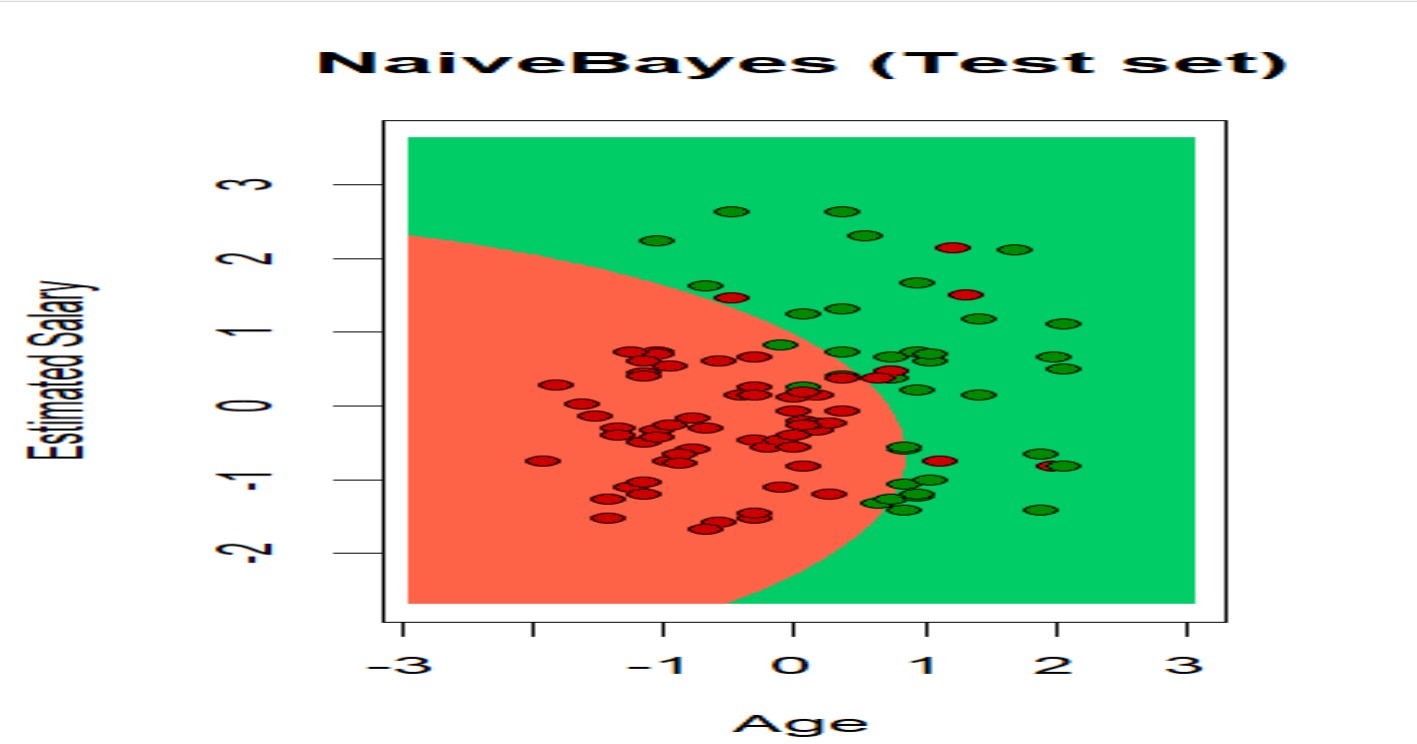
# Visualising the Test set results library(ElemStatLearn) set = test\_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01) X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01) grid\_set = expand.grid(X1, X2)

colnames(grid\_set) = c('Age', 'EstimatedSalary') y\_grid = predict(classifier, newdata = grid\_set)

plot(set[, -3], main = 'NaiveBayes (Test set)', xlab = 'Age', ylab = 'Estimated Salary', xlim = range(X1), ylim = range(X2)) contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE) points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'springgreen3', 'tomato')) points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))

**Output:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User ID | Gender | Age | EstimatedSalary | Purchased |
| 15624510 | Male | 19 | 19000 | 0 |
| 15810944 | Male | 35 | 20000 | 0 |
| 15668575 | Female | 26 | 43000 | 0 |
| 15603246 | Female | 27 | 57000 | 0 |
| 15804002 | Male | 19 | 76000 | 0 |
| 15728773 | Male | 27 | 58000 | 0 |
| 15598044 | Female | 27 | 84000 | 0 |
| 15694829 | Female | 32 | 150000 | 1 |
| 15600575 | Male | 25 | 33000 | 0 |
| 15727311 | Female | 35 | 65000 | 0 |
| 15570769 | Female | 26 | 80000 | 0 |
| 15606274 | Female | 26 | 52000 | 0 |
| 15746139 | Male | 20 | 86000 | 0 |
| 15704987 | Male | 32 | 18000 | 0 |
| 15628972 | Male | 18 | 82000 | 0 |

# Practical: 8

**Aim:** Text Analysis.

**Code:**

**Natural Language Processing**

dataset\_original = read.delim("C:\\Users\\Isha\\Downloads\\Restaurant reviews.csv", quote = '', stringsAsFactors = FALSE)

# Cleaning the texts install.packages('tm') install.packages('SnowballC') library(tm) library(SnowballC) corpus = VCorpus(VectorSource(dataset\_original$Review)) corpus = tm\_map(corpus, content\_transformer(tolower)) corpus = tm\_map(corpus, removeNumbers) corpus = tm\_map(corpus, removePunctuation) corpus = tm\_map(corpus, removeWords, stopwords()) corpus = tm\_map(corpus, stemDocument) corpus = tm\_map(corpus, stripWhitespace) # Creating the Bag of Words model dtm = DocumentTermMatrix(corpus) dtm = removeSparseTerms(dtm, 0.999) dataset = as.data.frame(as.matrix(dtm)) dataset$Liked = dataset\_original$Liked print(dataset$Liked)

# Encoding the target feature as factor dataset$Liked = factor(dataset$Liked, levels = c(0, 1)) # Splitting the dataset into the Training set and Test set install.packages('caTools') library(caTools) set.seed(123) split = sample.split(dataset$Liked, SplitRatio = 0.8) training\_set = subset(dataset, split == TRUE) test\_set = subset(dataset, split == FALSE)

# Fitting Random Forest Classification to the Training set install.packages('randomForest') library(randomForest) classifier = randomForest(x = training\_set[-692],

y = training\_set$Liked, ntree = 10)

# Predicting the Test set results y\_pred = predict(classifier, newdata = test\_set[-692])

# Making the Confusion Matrix cm = table(test\_set[, 692], y\_pred) print(cm)

**Output:**

> library(randomForest) randomForest 4.7-1.1

Type rfNews() to see new features/changes/bug fixes.

> classifier = randomForest(x = training\_set[-692], + y = training\_set$Liked,

+ ntree = 10)

> # Predicting the Test set results

> y\_pred = predict(classifier, newdata = test\_set[-692])

> # Making the Confusion Matrix

> cm = table(test\_set[, 692], y\_pred)

> print(cm)

y\_pred

0 1

1. 82 18
2. 23 77

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# Practical: 9

**Aim:** Classification model.

**Code:**

library(tree)

library(ISLR) library(rpart.plot) library(rattle) attach(Hitters) View(Hitters)

# Remove NA data

Hitters<-na.omit(Hitters)

# log transform Salary to make it a bit more normally distributed hist(Hitters$Salary)

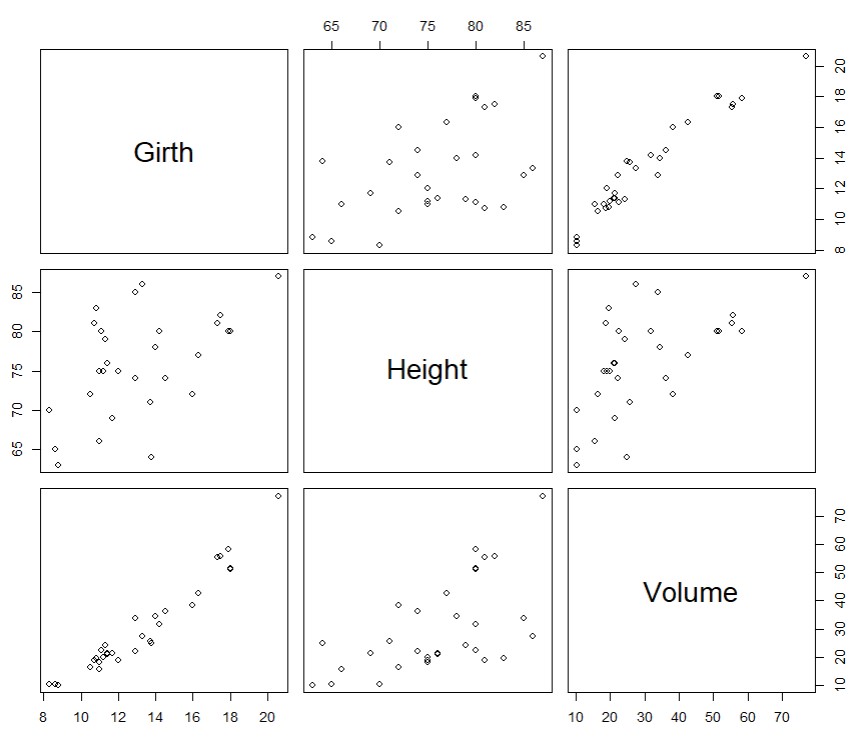
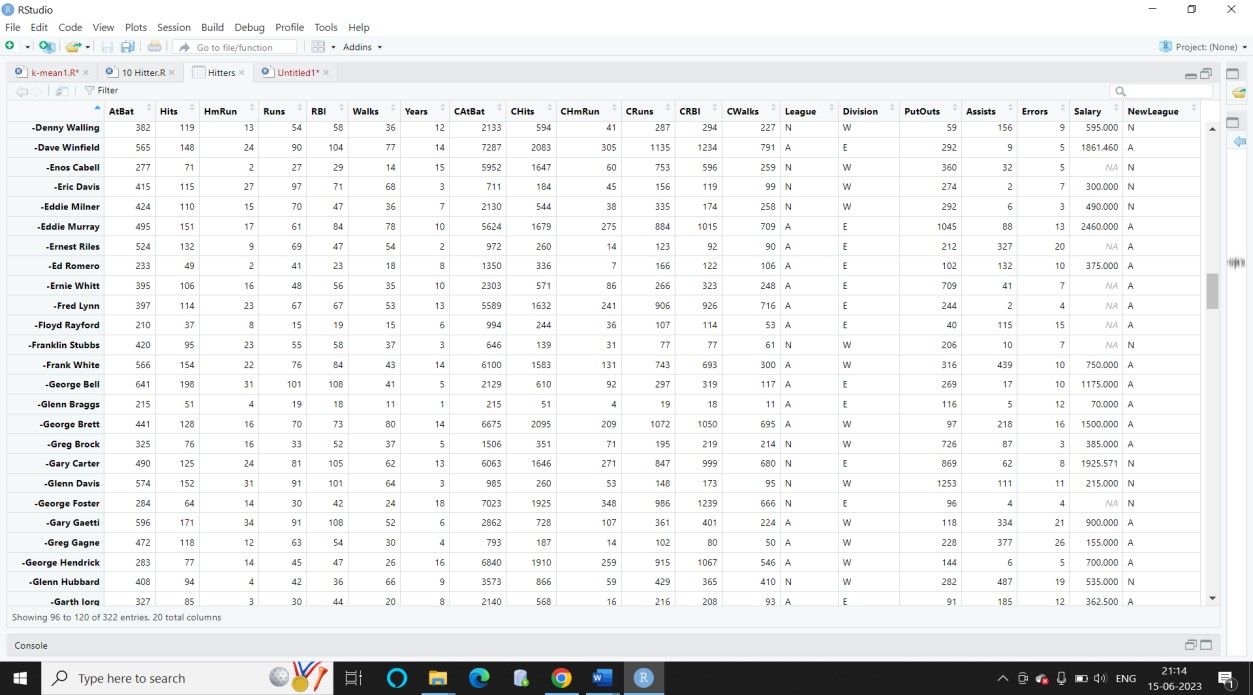
Hitters$Salary <- log(Hitters$Salary) hist(Hitters$Salary) tree.fit <- tree(Salary~Hits+Years, data=Hitters) summary(tree.fit) plot(tree.fit, uniform=TRUE,margin=0.2) text(tree.fit, use.n=TRUE, all=TRUE, cex=.8)

#plot(tree.fit) split <- createDataPartition(y=Hitters$Salary, p=0.5, list=FALSE) train <- Hitters[split,] test <- Hitters[-split,]

#Create tree model trees <- tree(Salary~., train) plot(trees) text(trees, pretty=0)

#Cross validate to see whether pruning the tree will improve performance cv.trees <- cv.tree(trees) plot(cv.trees) prune.trees <- prune.tree(trees, best=4) plot(prune.trees) text(prune.trees, pretty=0) yhat <- predict(prune.trees, test) plot(yhat, test$Salary) mean((yhat - test$Salary)^2)

**Output:**



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# Practical: 10 A

**Aim: Implement SVM Classification Technique.**

**Code:**

# Importing the dataset dataset = read.csv("C:\\Users\\Isha\\Downloads\\Social\_Network\_Ads.csv")

# Taking columns 3-5 dataset = dataset[3:5]

# Encoding the target feature as factor dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1)) # Splitting the dataset into the Training set and Test set install.packages('caTools') library(caTools) set.seed(123) split = sample.split(dataset$Purchased, SplitRatio = 0.75) training\_set = subset(dataset, split == TRUE) test\_set = subset(dataset, split == FALSE)

# Feature Scaling training\_set[-3] = scale(training\_set[-3]) test\_set[-3] = scale(test\_set[-3]) # Fitting SVM to the Training set install.packages('e1071') library(e1071)

classifier = svm(formula = Purchased ~ .,data = training\_set, type = 'C-classification',kernel = 'linear')

# Predicting the Test set results y\_pred = predict(classifier, newdata = test\_set[-3])

# Making the Confusion Matrix cm = table(test\_set[, 3], y\_pred)

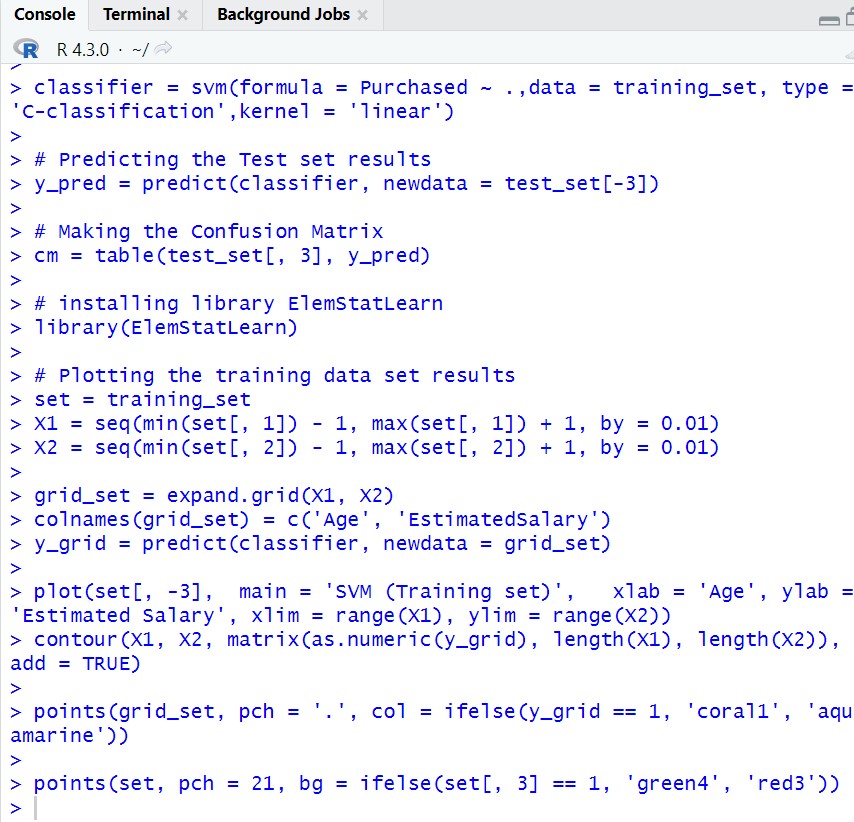
# installing library ElemStatLearn library(ElemStatLearn)

# Plotting the training data set results set = training\_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01) X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01) grid\_set = expand.grid(X1, X2) colnames(grid\_set) = c('Age', 'EstimatedSalary') y\_grid = predict(classifier, newdata = grid\_set)

plot(set[, -3], main = 'SVM (Training set)', xlab = 'Age', ylab = 'Estimated Salary', xlim = range(X1), ylim = range(X2)) contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE) points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'coral1', 'aquamarine')) points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))

**Output:**





**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

# Practical: 10 B

**Aim: Clustering Model.**

# Practical: 10 B

**Aim: Clustering Model.**

**Code:**

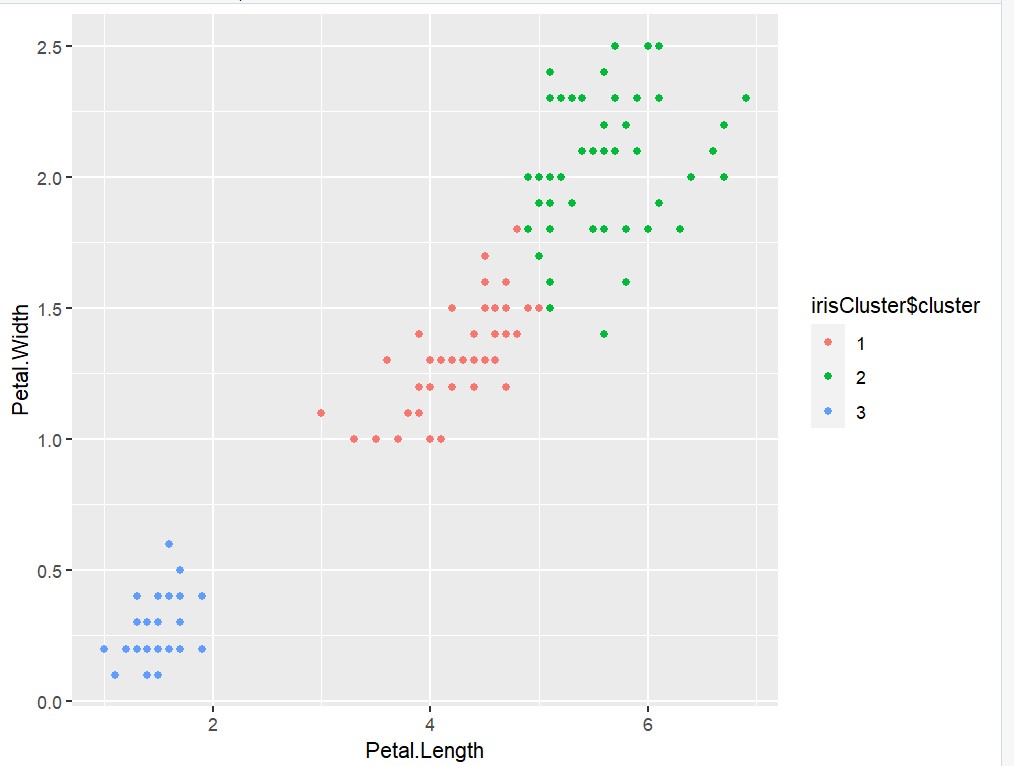
**i. Clustering algorithms for unsupervised classification.**

Install.packages(“ggplot2”) library(cluster)

library("ggplot2") set.seed(20) irisCluster <- kmeans(iris[, 3:4], 3, nstart = 20) irisCluster irisCluster$cluster <- as.factor(irisCluster$cluster) ggplot(iris, aes(Petal.Length, Petal.Width, color = irisCluster$cluster)) + geom\_point()

**Output:-**



 **ii. Plot the cluster data using R visualizations.**

rm(list=ls()) data = read.table("C:\\Users\\Isha\\Downloads\\food.txt", header=T,stringsAsFactors=F) mydata <- scale(data[,-1]) final = as.data.frame(cbind(data[,1],mydata)) head(final)

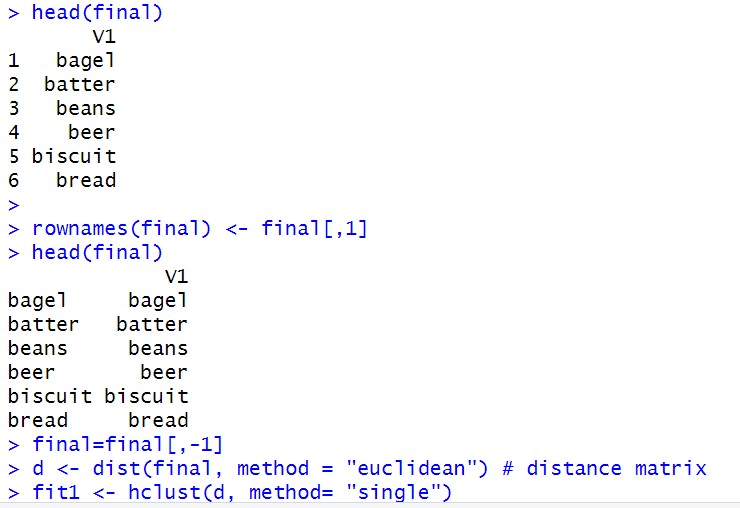
rownames(final) <- final[,1] head(final) final=final[,-1] d <- dist(final, method = "euclidean") # distance matrix fit1 <- hclust(d, method= "single") fit2 <- hclust(d, method= "complete") fit3 <- hclust(d, method= "average") fit4 <- hclust(d, method= "centroid")

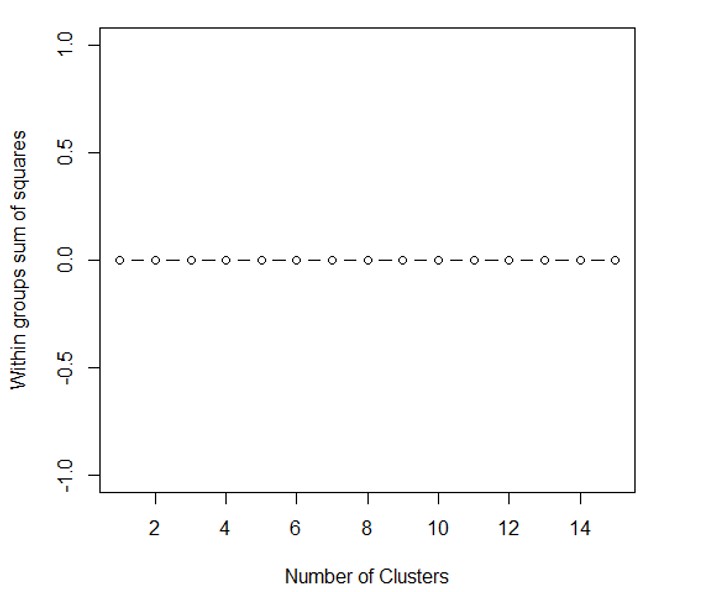
plot(fit1) # display dendogram plot(fit2) # display dendogram plot(fit3) # display dendogram plot(fit4) # display dendogram groups <- cutree(fit1, k=5) # cut tree into 5 clusters # draw dendogram with red borders around the 5 clusters rect.hclust(fit1, k=5, border="red")

wss <- rep(0, 15) wss[1] <- (nrow(final)-1)\*sum(apply(final,2,var)) for (i in 2:15) wss[i] <- sum(kmeans(final, centers=i)$withinss) plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares") fit5 <- kmeans(final, 5) # 5 cluster solution

food\_clusters <- data.frame(final, fit5$cluster)

**Output:**





## \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Practical Journal

**MODERN NETWORKING**

**A Practical Report**

Submitted in partial fulfillment of the

Requirements for the award of the Degree

**MASTER OF SCIENCE (INFORMATION TECHNOLOGY)**

**Submitted by**

**Ms. Sokhi Rinki Jassi**

**Seat No: 1312209**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

***COSMOPOLITAN'S***

**VALIA C.L. COLLEGE OF COMMERCE &**

**VALIA L.C. COLLEGE OF ARTS**

***(Affiliated to University of Mumbai)***

**D.N. NAGAR, ANDHERI (WEST), MUMBAI 400 053**