**1.Perform k means clustering on the IRIS dataset.plot wss to determine the optimumno of cluster to use.**

newiris=iris

head(newiris)

pairs(newiris)

newiris$Species = NULL

newiris[1:5,]kc=kmeans(newiris,3,10)

kc

kc$cluster

table(iris$Species, kc$cluster)

plot(newiris[c("Sepal.Length", "Sepal.Width")], col=kc$cluster

points(kc$centers[,c("Sepal.Length", "Sepal.Width")], col=1:3, pch=8, cex=3)

fviz\_cluster(kc,data=newiris)

k2=kmeans(newiris,2,25)

k3=kmeans(newiris,3,25)

k4=kmeans(newiris,4,25)

k5=kmeans(newiris,5,25)ggplot(aes(Sepal.Length, Sepal.Width, color = factor(cluster))) + geom\_text()

fviz\_cluster(k2,data=newiris)

p1 = fviz\_cluster(k2,data=newiris, geom='point') + ggtitle('k=2')

p2 = fviz\_cluster(k3,data=newiris, geom='point') + ggtitle('k=3')

p3 = fviz\_cluster(k4,data=newiris, geom='point') + ggtitle('k=4')

p4 = fviz\_cluster(k5,data=newiris, geom='point') + ggtitle('k=5')

library('gridExtra')

grid.arrange(p1,p2,p3,p4, nrow=2)

set.seed(125)

wss <- numeric(15)

wss

for (k in 1:15)

wss[k]=sum(kmeans(newiris,centers=k, nstart=25 ) $withinss )

wss

plot(1 : 15, wss, type="b", xlab="Number of Clusters " ,

ylab="Within Sum of Squares")

**2. Perform K-means clustering on the StudentGrades dataset. Plot WSS to**

**determine the optimum number of clusters to use**

install.packages('plyr')

install.packages('ggplot2')

install.packages('cluster')

install.packages('lattice')

install.packages('graphics')

install.packages('grid')

install.packages('gridExtra')

library (plyr)

library(ggplot2)

library(cluster)

library(lattice)

library(graphics)

library(grid)

library(gridExtra)

getwd()

setwd("D:/BDA")

grade\_input = read.csv('D:/BDA/grades\_km\_input.csv')

str(grade\_input)

kmdata = as.matrix(grade\_input[,2:4])

str(kmdata)

kmdata[1:5,]

wss = 15

for (k in 1:15)

wss[k]=sum(kmeans(kmdata,centers=k,nstart=25)$withinss)

wss

plot(

1:15,

wss,

type="b",

xlab = "No. of clusters",

ylab = "Withing Sum of Squares")

km = kmeans(kmdata,3, nstart=25)

km

df=grade\_input[,2:4]

str(df)

km$cluster

df$cluster = factor(km$cluster)

str(df$cluster)

centers = as.data.frame(km$centers)

centers

ggplot(data=df, aes(x=English, y=Math, color=cluster ))+

geom\_point() + theme(legend.position="right")

library('factoextra')

library('tidyverse')

library('ggplot2')

fviz\_cluster(km,data=kmdata, geom='point') + ggtitle('k=3')

g1= ggplot(data=df, aes(x=English, y=Math, color=cluster )) +

geom\_point() + theme(legend.position="right") +

geom\_point(data=centers,

aes(x=English,y=Math, color=as.factor(c(1,2,3))),

size=10, alpha=.3,show.legend = FALSE)

g2 =ggplot(data=df, aes(x=English, y=Science, color=cluster )) +

geom\_point () +

geom\_point(data=centers,

aes(x=English,y=Science, color=as.factor(c(1,2,3))),

size=10, alpha=.3,show.legend = FALSE)

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

geom\_point () +

geom\_point(data=centers,

aes(x=Math,y=Science, color=as.factor(c(1,2,3))),

size=10, alpha=0.3,show.legend = FALSE)

tmp = ggplot\_gtable(ggplot\_build(g1))

library('gridExtra')

grid.arrange(g1,g2,g3, nrow=3)

**3. Perform Hierarchical clustering on the Utilities dataset. Plot its dendogram &**

**Silhoutte Plot.**

utilities = read.csv("C:/clustering/utilities .csv")

str(utilities)

utilities

pairs(utilities[,2:8])

plot(Fuel\_Cost ~ Sales,utilities)

with(utilities,text(Fuel\_Cost ~ Sales,labels=Company, pos=1,cex=0.7))

utilities

z=utilities[,2:9]

z

m=apply(z,2,mean)

s=apply(z,2,sd)

m

s

z=scale(z,m,s)

distance = dist(z)

print(distance,digits=3)

hc.c=hclust(distance)

plot(hc.c)

plot(hc.c,labels = utilities$Company)

plot(hc.c,hang=-1)

hc.a=hclust(distance, method = "average")

plot(hc.a)

plot(hc.a,labels = utilities$Company)

plot(hc.a,hang=-1)

member.c=cutree(hc.c,3)

member.c

member.a=cutree(hc.a,3)

member.a

table(member.c, member.a)

aggregate(z, list(member.c), mean)

aggregate(utilities[,-c(1,1)], list(member.c), mean)

library(cluster)

plot(silhouette(cutree(hc.c,3),distance))

wss <- numeric(20)

for (k in 1:20)

wss[k]=sum(kmeans(z,centers=k, nstart=25 ) $withinss )

wss

plot(1 : 20, wss, type="b", xlab="Number of Clusters " ,

ylab="Within Sum of Squares" )

kc = kmeans(z,3)

kc

member.a

member.c

kc$cluster

plot(Sales ~ D.Demand, utilities)

plot(Sales ~ D.Demand, utilities, col =kc$cluster)

**4.Perform Association mining on the Groceries dataset. Generate frequent itemset**

**of size 1, 2, 3 & 4. Generate & Plot rules with support=0.001, confidence=0.6**

install.packages ( 'arules' )

install.packages ('arulesViz' )

library ( 'arules' )

library( 'arulesViz' )

data("Groceries")

View(Groceries)

summary(Groceries)

class(Groceries)

Groceries@itemInfo[1:20,]

Groceries@itemsetInfo[1:20,]

UDIT, MSc IT Sem II BIG DATA ANALYTICS JOURNAL (2021-22)

21

Manaswi M. Patil Seat No. 272011

g=Groceries@itemsetInfo[1:20,]

apply(Groceries@data[,1:30],2,

function(r) paste(Groceries@itemInfo[r,"labels"], collapse = ", "))

# Generating frequent itemset of size 1

itemsets = apriori(Groceries)

itemsets = apriori(Groceries, parameter =

list(minlen=1,

maxlen=1,

support=0.02,

target="frequent itemsets"))

inspect(itemsets)

UDIT, MSc IT Sem II BIG DATA ANALYTICS JOURNAL (2021-22)

22

Manaswi M. Patil Seat No. 272011

summary(itemsets)

UDIT, MSc IT Sem II BIG DATA ANALYTICS JOURNAL (2021-22)

23

Manaswi M. Patil Seat No. 272011

# The top 10 most frequent 1-itemsets

inspect(head(sort(itemsets, by = "support"),10))

# Generating frequent itemset of size 2

itemsets = apriori(Groceries, parameter =

list(minlen=2,

maxlen=2,

support=0.02,

target="frequent itemsets"))

inspect(itemsets)

summary(itemsets)

inspect(head(sort(itemsets, by = "support"),10))

itemsets = apriori(Groceries, parameter =

list(minlen=3,

maxlen=3,

support=0.02,

target="frequent itemsets"))

inspect(itemsets)

summary(itemsets)

inspect(head(sort(itemsets, by = "support"),10))

# Generating frequent itemset of size 4

itemsets = apriori(Groceries, parameter =

list(minlen=4,

maxlen=4,

support=0.02,

target="frequent itemsets"))

inspect(itemsets)

summary(itemsets)

rules = apriori(Groceries,

parameter=list(support=0.001,

confidence=0.6,

target="rules"))

inspect(rules)

summary(rules)

inspect(head(sort(rules, by = "support"),10))

# Rule Visualization

plot(rules)

plot(rules@quality)

inspect(head(sort(rules, by = "lift"),10))

confidentRules = rules[quality(rules)$confidence>0.9]

confidentRules

###

plot(confidentRules,

method="matrix",

measure = c("lift", "confidence"),

control=list(reorder=TRUE)

)

highLiftRules = head(sort (rules, by= "lift " ) , 5)

highLiftRules = head(sort(rules, by = "lift"))

highLiftRules

plot(highLiftRules, method = "graph", control = list(type="items"))

**5.create a bank sample data tree**

install.packages("rpart.plot")

library("rpart")

library("rpart.plot")

banktrain <- read.table("D:/BDA/bank-sample.csv",header=TRUE,sep=",")

banktrain=read.csv("D:/BDA/bank-sample.csv")

View(banktrain)

drops<-c("age", "balance", "day", "campaign", "pdays", "previous", "month")

banktrain <- banktrain [,!(names(banktrain) %in% drops)]

View(banktrain)

summary(banktrain)

fit <- rpart(subscribed ~ job + marital + education + default + housing + loan + contact +

poutcome,

method="class",

data=banktrain,

control=rpart.control(minsplit=1),

parms=list(split='information'))

summary(fit)

# Plot the tree

rpart.plot(fit, type=4, extra=2, clip.right.labs=FALSE, varlen=0, faclen=3)

fit <- rpart(subscribed ~ job + marital + education + default + housing + loan + contact +

duration + poutcome,

method="class",

data=banktrain,

control=rpart.control(minsplit=1),

parms=list(split='information'))

summary(fit)

rpart.plot(fit, type=4, extra=2, clip.right.labs=FALSE, varlen=0, faclen=3)

**6.** **create a decision tree for the bank sample dataset**

library("rpart") # load libraries

library("rpart.plot")

# Read the data

play\_decision <- read.csv("D:/BDA/DTdata.csv", header=TRUE, sep=",")

play\_decision

summary(play\_decision)

fit <- rpart(Play ~ Outlook + Temperature + Humidity + Wind,

method="class",

data=play\_decision,

control=rpart.control(minsplit=1),

parms=list(split='information'))

summary(fit)

?rpart.plot

rpart.plot(fit, type=4, extra=1)

rpart.plot(fit, type=4, extra=2, clip.right.labs=FALSE,

varlen=0, faclen=0)

newdata <- data.frame(Outlook="rainy", Temperature="mild",

Humidity="high", Wind=FALSE)

newdata

predict(fit,newdata=newdata,type="prob")

predict(fit,newdata=newdata,type="class")

**7. Simple linear regression**

income\_input = as.data.frame( read.csv("D:/BDA/income.csv"))

View(income\_input)

income\_input[1:10,]

summary(income\_input)

library(lattice)

splom(~income\_input[c(2:5)], groups=NULL, data=income\_input,

axis.line.tck = 0,

axis.text.alpha = 0)

pairs(income\_input[c(2:5)],)

results <- lm(Income~Age + Education + Gender, income\_input)

summary(results)

results2 <- lm(Income ~ Age + Education, income\_input)

summary(results2)

confint(results2, level = .95)

Age <- 41

Education <- 12

new\_pt <- data.frame(Age, Education)

new\_pt

conf\_int\_pt <- predict(results2, new\_pt, level=.95, interval="confidence")

conf\_int\_pt

pred\_int\_pt <- predict(results2, new\_pt, level=.95, interval="prediction")

pred\_int\_pt

**8. AIM:Logistic Regression**

churn\_input = as.data.frame(read.csv("D:/BDA/churn.csv"))

View(churn\_input)

head(churn\_input)

summary(churn\_input)

sum(churn\_input$Churned)

Churn\_logistic1 <- glm (Churned~Age + Married + Cust\_years + Churned\_contacts,

data=churn\_input, family=binomial(link="logit"))

summary(Churn\_logistic1)

Churn\_logistic2 <- glm (Churned~Age + Married + Churned\_contacts,

data=churn\_input, family=binomial(link="logit"))

summary(Churn\_logistic2)

Churn\_logistic3 <- glm (Churned~Age + Churned\_contacts,

data=churn\_input, family=binomial(link="logit"))

summary(Churn\_logistic3)

summary(Churn\_logistic2)

pchisq(.9 , 1, lower=FALSE)

# Receiver Operating Characteristic (ROC) Curve

install.packages("ROCR") #install, if necessary

library(ROCR)

pred = predict(Churn\_logistic3, type="response")

predObj = prediction(pred, churn\_input$Churned )

predObj

rocObj = performance(predObj, measure="tpr", x.measure="fpr")

aucObj = performance(predObj, measure="auc")

summary(aucObj)

plot(rocObj, main = paste("Area under the curve:", round(aucObj@y.values[[1]] ,4)))

**9.**

Open vmware > go on BDAVM >prac folder on vmware desktop>prac>go on 9 dots>type terminal and click on that you will seen pink window>minimize screen pink and go on documents icon<click on prac-pracs>click on that folder and quite>ctrl+l on address bar i.e home>remove /slash and udit press enter then you get hadoop and udit folder.>go on terminal for delete user> in pink screen type sudo deluser --remove-home Hadoop then press enter>add password as **udit@123>**go on addressbar and type home+enter you will show udit folder>type ctrl+c 3 times in pink window>enter command as sudo adduser hadoop+enter>new pas as **udit@123** again enter and type same password **udit@123** then enter full name as **h** enter **1+ enter 1 1 1+enter >**click on files then you will get created folder Hadoop then double click on hadoop folder then u will get 3 folders as name **.bashrc** **and .profile, .bashrlogout** then type **clear** in command window>type sudo nano sudoers then enter press and type pass as **udit@123+enter>**ctrl+x for save file and press y also.> enter command su – Hadoop+enter >password enter>enter command sudo apt update enter +password it will update>enter command as java -version to see jdk is install in our system or not/if not so sudo apt install default-jdk>known where we are for that we should follow go on desktop,prac,pracs and address bar ctrl+L enter>other location from icon vmware desktop>computer>usr double click>lib>jvm u can see all folders of java>java11 open jdk click on that then address bar ctrl+L and copy that path. And save it into a other place like notepad>clear command enter in pink window>open browser in vmware and type apache Hadoop 3.2.0 enter and click on 1st link which is visible click on that>then copy link from search bar>go on pink window and type command as wget [https://hadoop.apache.org/release/3.2.0.html /hadoop-3.2.0.tar.gz](https://hadoop.apache.org/release/3.2.0.html%20/hadoop-3.2.0.tar.gz) enter>type in command window ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa then enter> cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys then enter> chmod 0600 ~/.ssh/authorized\_keys then enter>type ssh localhost enter then type yes after question mark. Enter and clear the screen> $ wget [https://hadoop.apache.org/release/3.2.0.html/hadoop-3.2.0.tar.gz> then](https://hadoop.apache.org/release/3.2.0.html/hadoop-3.2.0.tar.gz%3e%20then) go on desktop ,prac,setup if not then go >downloads= hadoop-3.2.0.tar.gz>>address bar ctrl+L and /Hadoop and copy link and paste in window as sudo mv /home/udit/Downloads/Hadoop-3.2.0.tar.gz /home/Hadoop then enter password enter>tar xzvf Hadoop-3.2.0.tar.gz then> sudo nano .bashrc enter and go last of the file in window>go on other location>computer>home Hadoop>double click and copy path of Hadoop tar.gz version >go adrresbar and copy path

And paste into a notepad.

export JAVA\_HOME=/lib/jvm/java-11-openjdk-amd64

export PATH=$PATH:$JAVA\_HOME/bin

export HADOOP\_HOME=/home/hadoop/hadoop-3.2.0

export HADOOP\_COMMON\_HOME=/usr/local/hadoop/hadoop-3.2.0

export HADOOP\_MAPRED\_HOME=/usr/local/hadoop/hadoop-3.2.0

export PATH=$PATH:$HADOOP\_COMMON\_HOME/bin

export PATH=$PATH:$HADOOP\_COMMON\_HOME/Sbin enter and ctrl+x or s say yes

type in command wind as sudo nano .bashrc and source ~/.bashrc>cat .bashrc enter.>on clear window type java -version enter>Hadoop version>jps>sudo nano \$HADOOP\_HOME/etc/Hadoop/Hadoop-env.sh enter>cd $HADOOP\_HOME/ enter>ls>cd Hadoop>ls> enter>Hadoop-env.sh>sudo nano Hadoop-env.sh enter come on last line of code and type JAVA-HOME=/user/lib/jvm/java-11-openjdk-amd64/ ctrl xor s say yes enter>on command window>cd..>cd..>ls>cd bin>ls>Hadoop>jps>then computer,home,Hadoop,,Hadoop-3.2.0,etc,Hadoop>then command window type mkdir ~/input>cp $HADOOP\_HOME/etc/Hadoop/\*.xml ~/input>>then in command window type$HADOOP\_HOME/bin/Hadoop jar $HADOOP\_HOME/share/Hadoop/mapreduce/Hadoop-mapreduce-example-3.2.0.jar grep ~/grep\_example ‘allowed[.]\*’ then enter cat ~/grep\_example/\* enter>Hadoop version.

Practical : Install Hadoop in Pseudo Distributed Mode on Ubuntu

================================================================

Prerequisites

================================================================

An Ubuntu server VM with a user having sudo privileges

To get started, we’ll update our package list:

sudo apt update

Next, we’ll install OpenJDK, the default Java Development Kit on Ubuntu 18.04:

sudo apt install default-jdk

Once the installation is complete, let’s check the version.

java -version

This output verifies that OpenJDK has been successfully installed.

Create a new user- hadoop

sudo adduser hadoop

Give all priviliges to hadoop

sudo nano /etc/sudoers

Add the line

user\_name ALL=(ALL:ALL) ALL

switch user to hadoop15

su - hadoop

The prompt shoud look like this - hadoop@ubuntu:/

It is also required to set up key-based ssh

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

chmod 0600 ~/.ssh/authorized\_keys

verify key based login

ssh localhost

=========================================================

Step 2 — Downloading & Installing Hadoop

Visit the Apache Hadoop Releases page to find the most recent stable release.

https://hadoop.apache.org/release/3.2.0.html

we’ll install Hadoop 3.2.0.

There are 2 options to download and install hadoop

1. Download and install using one single command

wget https://hadoop.apache.org/release/3.2.0.html/hadoop-3.2.0.tar.gz

This will download the mentioned version and then install it on your system

2. If the above does not work then go to the webpage of Apache Hadoop, download it manually and then install it

visit website

https://hadoop.apache.org/release/3.2.0.html

Click button that says "Download tar.gz" (download the latest version)

install hadoop using the following command

we’ll use the tar command with the -x flag to extract, -z to uncompress, -v for verbose output, and -f to specify that we’re extracting from a file.

tar xzvf hadoop-3.2.0.tar.gz

Finally, we’ll move the extracted files into /usr/local, the appropriate place for locally installed software.

sudo mv hadoop-3.2.0 /home/hadoop

verify by navigating to /home/hadoop

===================================================================================

Update 6 important files

===================================================================================

1. .bashrc

===================================================================================

Set path for Java & Hadoop

sudo nano .bashrc

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64

export PATH=$PATH:$JAVA\_HOME/bin

export HADOOP\_HOME = /home/hadoop15/hadoop-3.2.0

export HADOOP\_INSTALL = $HADOOP\_HOME

export HADOOP\_COMMON\_HOME = $HADOOP\_HOME

export HADOOP\_MAPRED\_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

execute by following line

source ~/.bashrc

Checking of java and hadoop

Command: java -version

Command: hadoop version

=================================================================================

2 hadoop-env.sh - Configuring Hadoop’s Java Home

=================================================================================

To Configure Hadoop15’s Java Home, begin by opening hadoop-env.sh

sudo nano /home/hadoop/hadoop-3.2.0/etc/hadoop/hadoop-env.sh

or use the variable $HADOOP\_HOME

sudo nano $HADOOP\_HOME/etc/hadoop/hadoop-env.sh

Add the following line at the end of .sh file

export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64

=================================================================================

3 core-site.xml

=================================================================================

sudo nano $HADOOP\_HOME/etc/hadoop/core-site.xml

<property>

<name>hadoop.tmp.dir</name>

<value>/home/hadoop/tmpdata</value>

<description>A base for other temporarary directories</description>

</property>

<property>

<name>fs.default.name</name>

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

<description>The name of the default file system.</description>

</property>

=================================================================================

4 hdfs-site.xml

=================================================================================

sudo nano $HADOOP\_HOME/etc/hadoop/hdfs-site.xml

<property>

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

<value>/home/hadoop/dfsdata/namenode</value>

<description>Location of namenode</description>

</property>

<property>

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

<value>/home/hadoop/dfsdata/datanode</value>

<description>Location of datanode</description>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

<description>Replication Factor</description>

</property>

=================================================================================

5 mapred-site.xml

=================================================================================

sudo nano $HADOOP\_HOME/etc/hadoop/mapred-site.xml

<property>

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

<value>yarn</value>

<description>Name of my mapreduce framework</description>

</property>

=================================================================================

6 yarn-site.xml

=================================================================================

sudo nano $HADOOP\_HOME/etc/hadoop/yarn-site.xml

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

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

</property>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>127.0.0.1</value>

</property>

<property>

<name>yarn.acl.enable</name>

<value>0</value>

</property>

<property>

<name>yarn.nodemanager.env-whitelist</name>

<value>JAVA\_HOME, HADOOP\_COMMON\_HOME, HADOOP\_HDFS, HADOOP\_YARN</value>

</property>

=================================================================================

Format Namenode

=================================================================================

move to the /bin folder

cd /home/hadoop/hadoop-3.2.0/bin

Now format the namenode using the following command

hdfs namenode -format

=================================================================================

Start the namenode, datanode

=================================================================================

move to the /sbin folder

/home/hadoop/hadoop-3.2.0/sbin

start-dfs.sh

start-yarn.sh

=================================================================================

Start the task tracker and job tracker // Skip in case file not present

=================================================================================

start-mapred.sh

=================================================================================

To check if Hadoop started correctly

=================================================================================

jps

=================================================================================

Copy file from local system to hdfs

=================================================================================

hdfs dfs -put source

check using ls

hdfs dfs -ls

=================================================================================

Check with browser

=================================================================================

DFS overview - http://localhost:9870/

Datanode - http://localhost:9864/