# **INTRODUCTION**

**NAME - PARAS PANT**

**COURSE - B. SC. COMPUTER SCIENCE (HON.)**

**SECTION - B**

**SEMESTER - 6th**

**COLLEGE ROLL NO. - 7848**

**EXAM ROLL NO. - 18035570054**

**EMAIL ID -** [**paraspat09@gmail.com**](mailto:paraspat09@gmail.com)

**CONTACT - 9667602924**

**DATA MINING PRACTICAL FILE**

# INDEX

1. Create a file “people.txt”
   1. Read the data from the file
   2. Create a rule set E that contain rules to check for the following conditions
      1. The age should be in the range 0-150.
      2. The age should be greater than years married.
      3. The status should be married or single or widowed.
      4. If age is less than 18 the age group should be child, if age is between 18 and 65 the age group should be adult, if age is more than 65 the age group should be elderly.
   3. Check whether rule set E is violated by the data in the file people.txt.
   4. Summarize the results obtained
   5. Visualize the results obtained
   6. Code
   7. Screenshots
2. Perform the following preprocessing tasks on the dirty\_iris dataset.
   1. Calculate the number and percentage of observations that are complete.
   2. Replace all the special values in data with NA.
   3. Define these rules in a separate text file and read them.(Use editfile function in R (package editrules).
   4. Print the resulting constraint object.
      1. Species should be one of the following values: setosa, versicolor or virginica.
      2. All measured numerical properties of an iris should be positive.
      3. The petal length of an iris is at least 2 times its petal width.
      4. The sepal length of an iris cannot exceed 30cm.
      5. The sepals of an iris are longer than its petals.
   5. Determine how often each rule is broken (violatedEdits). Also summarize and plot the result.
   6. Find outliers in sepal length using box plot and boxplot.stats
   7. Code
   8. Screenshots
3. Preprocess
   1. Iris Dataset
      1. Scale
      2. Center
   2. Wine Dataset
      1. Scale
      2. Center
   3. Code
   4. Screenshots
4. Run Apriori algorithm to find frequent itemsets and association rules
   1. Use minimum support as 50% and minimum confidence as 75%
   2. Use minimum support as 60% and minimum confidence as 60%
   3. Code
   4. Screenshots
5. Use Naive bayes, K-nearest, and Decision tree classification algorithms and build classifiers. Divide the data set into training and test sets. Compare the accuracy of the different classifiers under the following situations
   1. Training and Testing set division
      1. Training set = 75% Test set = 25%
      2. Training set = 66.6% (2/3rd of total), Test set = 33.3%
   2. Training set is chosen by
      1. hold out method
      2. Random subsampling
      3. Cross-Validation.
      4. Compare the accuracy of the classifiers obtained.
   3. Data is scaled to standard format.
   4. Code
   5. Screenshots
6. Use Simple K Means, DBScan, Hierarchical clustering algorithms for clustering. Compare the performance of clusters by changing the parameters involved in the algorithms.
   1. Code
   2. Screenshots

# QUESTION 1

**Program 1 : Create a file “people.txt” with the following data:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Age** | **Age Group** | **Height** | **Status** | **Years Married** |
| 21 | adult | 6.0 | single | -1 |
| 2 | child | 3 | married | 0 |
| 18 | adult | 5.7 | married | 20 |
| 221 | elderly | 5 | widowed | 2 |
| 34 | child | -7 | married | 3 |

**i) Read the data from the file “people.txt”.**

**ii) Create a rule set E that contain rules to check for the following conditions :**

1. **The age should be in the range 0-150.**
2. **The age should be greater than years married.**
3. **The status should be married or single or widowed.**
4. **If age is less than 18 the age group should be child, if age is between 18 and 65 the age group should be adult, if age is more than 65 the age group should be elderly.**

**iii) Check whether rule set E is violated by the data in the file people.txt.**

**iv) Summarize the results obtained in part(iii)**

**v) Visualize the results obtained in part(iii)**

library(editrules)

people=read.table("C:/Users/Girish/Desktop/paras/DM/Q1Table.txt",TRUE,"\t")

people

E=editset(expression(

Age>=0,

Age<=150,

Age>yearsmarried,

status %in% c("married","single","widowed"),

if(Age<18) agegroup == "child",

if(Age>=18 && Age<65) agegroup == "adult",

if(Age>=65) agegroup == "elder"

))

E

ve=violatedEdits(E,people)

ve

summary(ve)

plot(E)

plot(ve)

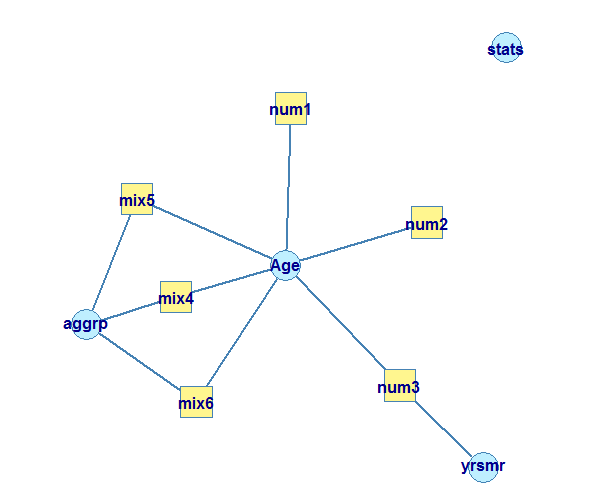
plot(E,layout=layout.circle)

> 
people 
> 
Age agegroup height 
1 
3.0 
5.7 
4 
5.0 
peopl e=read. tabl /User s/Gi ri sh/oesktop/paras/DM/QITab1 e. txt" , TRUE , 
21 
18 
221 
34 
adult 
child 
adult 
elderly 
child 
6. 
-7. 
O 
O 
status yearsmarried 
si ngle 
marri ed 
married 
wi dowed 
marri ed 

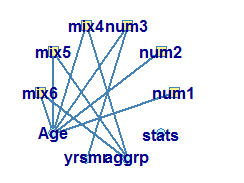
Data 
dat5 
dat6 . 
Edit 
num1 : 
num2 
num3 
mi x4 
mi x5 
mi x6 
model : 
agegroup %in% adult' , 
status c('married', 'single', 
set: 
O Age 
. Age 150 
. yearsmarried < Age 
'child', 
'elder 
' wi dowed ' ) 
: Age < 18 ) ! ( agegroup c('adult', 
: 18 Age & Age < 65 ) ! ( 
agegroup %in% 
: 65 Age ) ! ( agegroup adult', 
'elder') ) 
c('child', 
'child') ) 
'elder') ) 

ed1T 
num1 
record 
1 
4 
num2 
Τ2Ι_.ΙΕ 
num3 
TRI_IE 
τΙλΤ5 
TRl_lE 
daT6 
mfx5 
Τ2Ι_.ΙΕ 

> summary(ve) 
Edit violations, 5 observations, O completely missing (0%): 
editname freq 
rel 
num2 
num3 
dati 
mix5 
1 
1 
1 
1 
Edit violations per record: 
errors freq 
1 
rel 



mix6 
mix4 
dat6 
Z numl 
mix5 
dat5 
num3 
num2 
o oo 
10 
005 
1.2 
Edit violation frequency of top 8 edits 
0.10 
Frequency 
Edit violations per record 
2 records with no violations 
1.6 
Number of violations 
0.15 
1.8 
0.20 
20 



# QUESTION 2

**Program 2 : Perform the following preprocessing tasks on the dirty\_iris dataset.**

**i) Calculate the number and percentage of observations that are complete.**

**ii) Replace all the special values in data with NA.**

**iii) Define these rules in a separate text file and read them.**

**(Use editfile function in R (package editrules).**

**Print the resulting constraint object.**

**– Species should be one of the following values: setosa, versicolor or virginica.**

**– All measured numerical properties of an iris should be positive.**

**– The petal length of an iris is at least 2 times its petal width.**

**– The sepal length of an iris cannot exceed 30cm.**

**– The sepals of an iris are longer than its petals.**

**iv) Determine how often each rule is broken (violatedEdits). Also summarize and plot the result.**

**v) Find outliers in sepal length using box plot and boxplot.stats**

library(editrules)

dirty\_iris=read.csv("https://raw.githubusercontent.com/edwindj/datacleaning/master/data/dirty\_iris.csv")

dirty\_iris

cmplt\_cases=complete.cases(dirty\_iris)

as.numeric(cmplt\_cases)

sum(as.numeric(cmplt\_cases))

num\_of\_cmplt\_obsrvn=sum(cmplt\_cases)

num\_of\_obsrvn=nrow(dirty\_iris)

percentage=(num\_of\_cmplt\_obsrvn/num\_of\_obsrvn)\*100

sapply(dirty\_iris,is.infinite)

sapply(dirty\_iris,is.nan)

dirty\_iris[sapply(dirty\_iris,is.infinite)]=NA

dirty\_iris[sapply(dirty\_iris,is.nan)]=NA

E=editfile("C:/Users/Girish/Desktop/paras/DM/Q2Rules.txt","all")

E

ve=violatedEdits(E,dirty\_iris)

ve

summary(ve)

plot(E)

plot(ve)

plot(E,layout=layout.circle)

sepal\_length=dirty\_iris$Sepal.Length

hist(sepal\_length)

boxplot(sepal\_length,horizontal = T)

boxplot.stats(sepal\_length)

new\_sepal\_length=sepal\_length[sepal\_length<20]

new\_sepal\_length=new\_sepal\_length[new\_sepal\_length>4]

boxplot(new\_sepal\_length,horizontal = T)

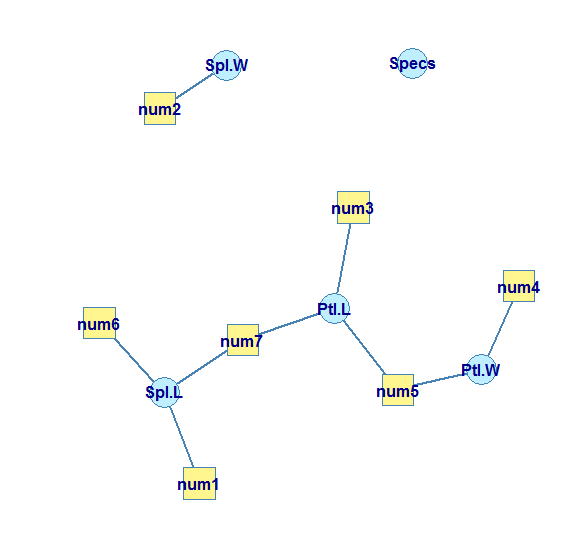
boxplot.stats(new\_sepal\_length)

hist(new\_sepal\_length)

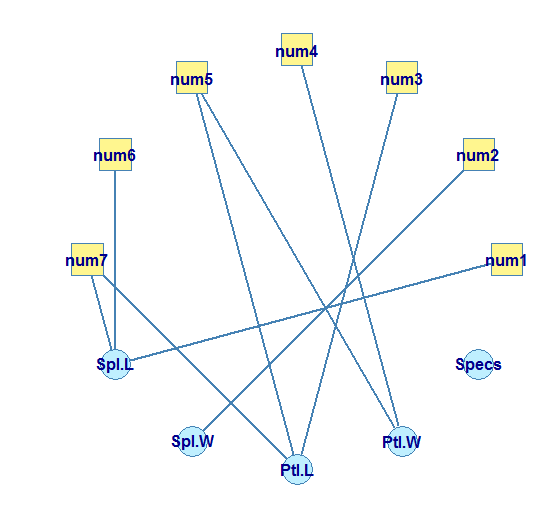
> cases (di rty_i ris) 
> as. numeri 
191] 1 01110101110 
o 
o 
1 
o 
o 
1010 
1110 
0101 
1001 
1001 
1 
1 
1 
o 
o 
o 
1 
1 
o 
1 
o 
1 
1 
1 
1 
o 
1 
1 
o 
o 
o 
o 
1 
1 
o 
o 
1 
1 
1 
1 
1 
1 
1 
o 
1 
o 
1 
1 
1 
1 
1 
1 
1 
1 
1 
1 
1 
1 
o 
1 
o 
O 
1 
1 
1 
1 
o 
1 
O 
O 
O 
1 
1 
O 
O 
> sum(as. 
96 
> rty_iris) 
96 
> num_of_obsrvn 
150 
> percentage 
64 

Data 
d atl 
Edit 
numl 
num2 
num3 
num4 
num5 
num6 
num7 . 
model : 
species %in% c('setosa' , 
set: 
: O sepal . Length 
: O sepal. width 
: O petal. Length 
: O petal. width 
'versicolor' , 
'virginica') 
2"peta1.width petal . Length 
sepal . Length 30 
petal . Length < sepal. Length 

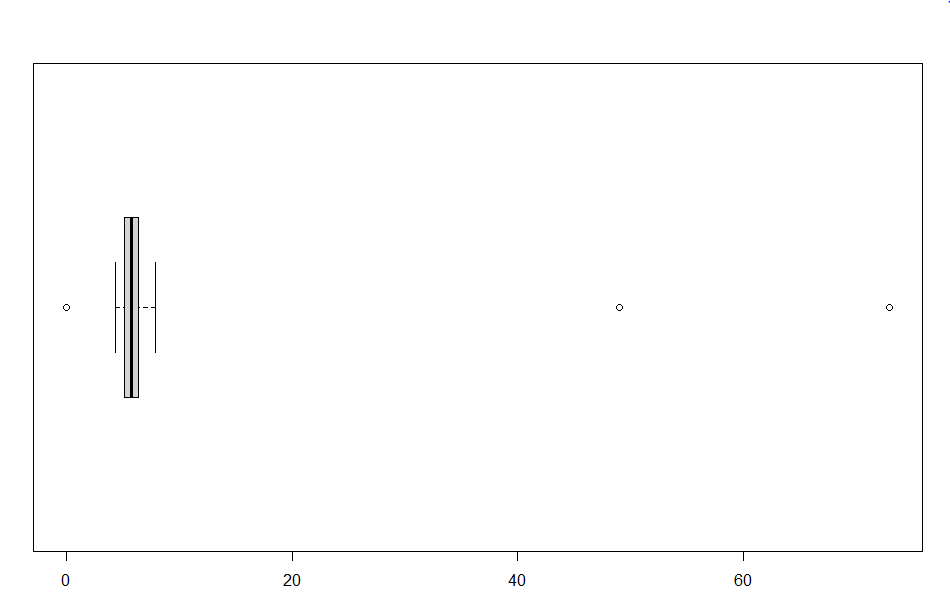
> summary(ve) 
Edit violations, 150 observations , 
editname freq rel 
60. 7% 
11. 
O completely missing 
nums 
num6 
num7 
num2 
2 1.3% 
2 1.3% 
1 0.7% 
Edit violations per record: 
errors freq 
o 
1 
91 
17 
13 
4 
1 
rel 
8.7% 
0.7% 



datl 
num4 
num3 
Z numl 
num2 
num7 
num6 
num5 
0 000 
O 
0 005 
2 
Edit violation frequency of top 8 edits 
0010 
Frequency 
Edit violations per record 
91 records with no violations 
3 
Number of violations 
0015 
4 
0 020 
5 



20 
Histogram of sepal_length 
40 
sepal length 
60 
80 



> boxplot. stats (sepal_length) 
Sstats 
4.30 5.10 5.75 6.40 7.90 
140 
S conf 
5. 576405 5.923595 
Sout 
73 0 49 

4.5 
50 
5.5 
60 
6.5 
70 
7.5 
80 

> boxplot. 
Sstats 
4.3 5.1 5.7 6.4 7.9 
Sn 
137 
Sconf 
[1 ) 5. 524515 5.875485 
Sout 
numeri c (O) 

Histogram of new_sepal_length 
4 
sepal_length 
5 
new 
6 
7 
8 

# QUESTION 3

**Program 3 : Load the data from the wine dataset. Check whether all attributes are standardized or not (mean is 0 and standard deviation is 1). If not, standardize the attributes. Do the same with Iris dataset.**

**Run following algorithms on 2 real datasets and use appropriate evaluation measures to compute correctness of obtained patterns:**

###iris Dataset

iris

summary(iris)

#mean(iris$Sepal.Length)

sapply(iris[,-5],sd)

#install.packages("caret",dependencies = TRUE)

library(caret)

#preProcess()

#"center" subtracts the mean,"scale" divides by the standard deviation

preprocessValue=preProcess(iris[,-5],method = c("scale","center"))

iris\_transformed=predict(preprocessValue,iris[,1:4])

sapply(iris\_transformed,sd)

summary(iris\_transformed)

###Wine Dataset

wine=read.csv("C:/Users/Girish/Desktop/paras/DM/wine.csv")

summary(wine)

sapply(wine[,-1],sd)

#preProcess()

preprocessValue=preProcess(wine[,-1],method = c("scale","center"))

wine\_transformed=predict(preprocessValue,wine[,2:14])

sapply(wine\_transformed,sd)

summary(wine\_transformed)

> summary(iris) 
sepal . Length 
sepal . wi dth 
petal . Length 
petal . wi dth 
1st Qu 300 
Med i an . 300 
3rd Qu. 300 
3rd Qu. :5.100 
3rd Qu 800 
Min. 
300 
1st Qu. 
100 
Medi an 
800 
:5.843 
Mean 
3rd Qu 
. :6.400 
900 
speci es 
setosa 
versi color : 50 
virginica : 50 
Min. 
000 
1st Qu. 
800 
Median : 
3. 000 
.3. 057 
Mean 
. 400 
Max. 
Min. 
000 
1st Qu. 
600 
Medi an . 
350 
3.758 
Mean 
900 
Max. 
Min. 
Mean 
Max. 
:o.100 
.:1. 
500 
> . Length) 
> sapp1y(irisC, 
-51 ,sd) 
sepal . Length 
o. 8280661 
sepal . width petal . Length 
0.4358663 
1. 7652982 
petal . width 
o. 7622377 

> ,sd) 
sepal . Length sepal . wi dth petal . Length petal . width 
. :-1.2225 
petal . width 
sepal . Length 
Min. 
:-1. 86378 
1st Qu. : 
-o. 89767 
Median : 
-o. 05233 
: o. 00000 
Mean 
3rd Qu. : 
o. 67225 
. 2.48370 
Max. 
Min. 
: -1. 4422 
1st Qu. : 
-1. 1799 
Median . 
• 0.1321 
: o. 0000 
Mean 
3rd Qu. : 
o. 7880 
: 1.7064 
sepal . width 
Min. 
:-2.4258 
1st Qu. : 
-o. 5904 
Median • 
. -o. 1315 
: o. 0000 
Mean 
3rd Qu. : 
o. 5567 
3. 0805 
Max. 
petal . Length 
Min. 
:-1. 5623 
1st Qu 
Median : 
o. 3354 
: o. 0000 
Mean 
3rd Qu. : 
o. 7602 
Max. 

> summary(wine) 
Malic. acid 
:11. 03 
1st Qu. 210 
Medi an . 360 
:13.05 
367 
3rd Qu. 000 
3rd Qu. 558 
:14. 83 
88.00 
99. 74 
: 1.280 
1st Qu. 938 
Median :2.780 
: 4.690 
:13. 000 
Type 
Min. 
1st Qu. 
Median : 
Mean 
Max. 
A1 cohol 
Ash 
000 
000 
2. 000 
938 
B . 000 
:10. 60 
:17. 20 
:19. 50 
:19.49 
:21. 50 
: 30. oo 
Min. 
1st Qu 
Medi an 
Mean 
3rd Qu 
Max. 
. 36 
OO 
. :13. 68 
Min. 
1st Qu. 
Medi an 
Mean 
3rd Qu. 
740 
603 
865 
336 
:3.083 
800 
Min. 
Mean 
Max. 
360 
3. 230 
A1 cal inity. of. ash Magnes i um 
Total . phenols 
Min. 
1st Qu. 
Medi an 
Mean 
3rd Qu. 
Min. 
1st Qu. : 
Median : 
Mean 
3rd Qu 
. 70 
98. 
. :107. 
:162. 
. OO 
OO 
OO 
OO 
Min. 
980 
1st Qu. 742 
Median 355 
Mean 
3rd Qu. 
Max. 
:2.295 
800 
:3.880 
F 1 avanoi ds 
Nonflavanoid. phenols proanthocyanins 
Min. 
1st Qu. 
Medi an 
Mean 
3rd Qu. 
Max. 
340 
205 
:2.135 
029 
875 
080 
Min. 
:o.1300 
1st Qu 
. 2700 
Medi an . 
3400 
3619 
Mean 
3rd Qu 
. :0.4375 
6600 
Max. 
Min. 
:o.410 
1st Qu. :1.250 
Median :1. 555 
Mean 
3rd Qu 
Max. 
591 
. 950 
:3.580 
col or. intensity 
00280.00315. of. di luted. wi nes 
Min. 
1st Qu. : 
Medi an 
Mean 
3rd Qu. : 
3.220 
5.058 
6. 200 
Min. 
1st Qu 
Medi an 
Mean 
3rd Qu 
Max. 
4800 
. 7825 
9650 
9574 
. 1200 
7100 
Min. 
Mean 
3rd Qu. : 
Max. 
270 
2. 612 
3.170 
000 
prol ine 
Min. 
. 278.0 
1st Qu. : 
500. 5 
Median : 673 
: 746.9 
Mean 
3rd Qu. : 
985.0 
:1680. o 

> ,sd) 
00280. 00315. 
Al cohol 
o. 8118265 
As h 
o. 2743440 
Magnes i um 
14. 2824835 
Fl avanoids 
o. 9988587 
Pr oanthocyani ns 
o. 5723589 
Hue 
o. 2285716 
proline 
314. 9074743 
Malic. acid 
1. 1171461 
Alcalinity. of. ash 
3. 3395638 
Total . phenol s 
o. 6258510 
Nonfl avanoi d. pheno s 
o. 1244533 
color. intens icy 
2. 3182859 
of. di luted. wi nes 
o. 7099904 

> ,sd) 
A 1 cohol 
Ash 
M agnes i um 
Fl avanoi ds 
Pr oant hocyani ns 
Malic. acid 
Al cal inity. of. ash 
Total . phenol s 
Nonfl avanoi d. phenol s 
color. intensi ty 
Hue 00280.00315. of. diluted. wines 
prol i ne 

A1 cohol 
Min. 
2.42739 
1st Qu. : 
-o. 78603 
Median . 
• o. 06083 
: o. 00000 
Mean 
3rd Qu. : 
o. 83378 
2. 25341 
Max. 
A1 cal inity.of. ash 
Min. 
2. 663505 
1st Qu. : 
-o. 687199 
Median . 
• o. 001514 
: o. 000000 
Mean 
3rd Qu • 
o. 600395 
3. 145637 
F 1 avanoids 
Min. 
: -1. 6912 
1st Qu. : 
-o. 8252 
Median . 
• 0.1059 
: o. 0000 
Mean 
3rd Qu. : 
o. 8467 
3. 0542 
Max. 
col or. intensity 
Min. 
:-1. 6297 
1st Qu. : 
-o. 7929 
Median . 
• -o. 1588 
: o. 0000 
Mean 
3rd Qu. : 
0.4926 
3.4258 
Max. 
Mal ic. acid 
Min. 
. -1. 4290 
1st Qu. : 
-o. 6569 
Median : 
-o. 4219 
: o. 0000 
Mea n 
3rd Qu. : 
o. 6679 
3. 1004 
Magnes um 
Min. 
1st Qu. 
Medi an 
Mea n 
3rd Qu. 
Max. 
: -2. 0824 
-o. 8221 
-0.1219 
o. 0000 
o. 5082 
4. 3591 
Ash 
Min. 
3. 66881 
1st Qu • 
. .-o. 57051 
Median : 
-o. 02375 
: o. 00000 
Mean 
3rd Qu • 
o. 69615 
3.14745 
Max. 
Total . phenols 
Min. 
. -2. 10132 
1st Qu. : 
-o. 88298 
Median : 
o. 09569 
: o. 00000 
Mean 
3rd Qu. : 
o. 80672 
. 2.53237 
Max. 
Nonflavanoid. phenols proanthocyanins 
Min. 
. -1. 8630 
1st Qu. : 
-o. 7381 
Median : 
-0.1756 
: o. 0000 
Y earl 
3rd Qu.: 
o. 6078 
2. 3956 
Min. 
2. 08884 
1st Qu. : 
-o. 76540 
Median : 
o. 03303 
: o. 00000 
Y earl 
3rd Qu.: 
o. 71116 
3. 29241 
Min. 
2. 06321 
1st Qu.: 
-o. 59560 
Medi an 
:-o. 06272 
: o. 00000 
Mean 
3rd Qu.: 
o. 62741 
: 3.47527 
Max. 
00280.00315. of. di luted. wines 
Min. 
:-1. 8897 
1st Qu. : 
-o. 9496 
Median . 
• 0.2371 
: o. 0000 
Mean 
3rd Qu. : 
o. 7864 
: 1.9554 
prol ine 
Min. 
: -1. 4890 
1st Qu. : 
-o. 7824 
Median : 
-o. 2331 
: o. 0000 
Mean 
3rd Qu.: 
o. 7561 
2. 9631 
Max. 

# QUESTION 4

**Program 4 : Run Apriori algorithm to find frequent itemsets and association rules**

**4.1 Use minimum support as 50% and minimum confidence as 75%**

**4.2 Use minimum support as 60% and minimum confidence as 60%**

#install.packages("arules")

library(arules)

data("Groceries")

?Groceries

str(Groceries)

head(Groceries)

inspect(head(Groceries))

summary(Groceries)

itemFrequencyPlot(Groceries,topN=20,type="absolute")

#Most Popular=Whole milk,other vegetables

#Not so popular=brown bread, domestic eggs

rules=apriori(Groceries,parameter = list(supp=0.001,conf=0.8))

rules=apriori(Groceries,parameter = list(supp=0.5,conf=0.75)) #0 rules in it

rules=apriori(Groceries,parameter = list(supp=0.6,conf=0.6)) #0 rules in it

inspect(rules)

head(rules)

inspect(head(rules))

inspect(rules[1:10])

rules=sort(rules,by="confidence",decreasing = T)

#install.packages("arulesViz")

library(arulesViz)

library(plyr)

library(dplyr)

plot(rules[1:5],method="graph",engine='interactive',shading=NA)

#Bakery Dataset

bakery=read.csv("C:/Users/Girish/Desktop/paras/DM/BakeryDataSet/1000/1000i.csv")

#will not work

bakeryrules=apriori(bakery,parameter = list(supp=0.001,conf=0.8))

#this method will work

names(bakery) <- c("Receipt\_Number","Food","Quantity") #add column names

id<-c(1:5)

food<-c("milk","sugar","chocolate","apples","curd")

df <- data.frame(id, food)

bakery$Food=df$food[bakery$Food]

bakery=ddply(bakery,"Receipt\_Number",function(bakery) paste(bakery$Food,collapse = ','))

bakery$Receipt\_Number=NULL

names(bakery) <- c("item")

write.csv(bakery, "C:/Users/Girish/Desktop/paras/DM/tmp/1000transac.csv",quote = FALSE,row.names = FALSE)

bakeryTransacs <- read.transactions(

file = "C:/Users/Girish/Desktop/paras/DM/tmp/1000transac.csv",

format = "basket",

sep = ",",

header = TRUE

)

inspect(head(bakeryTransacs))

bakeryrules=apriori(bakeryTransacs,parameter = list(supp=0.002,conf=0.08))

inspect(head(bakeryrules))

plot(bakeryrules[1:5],method="graph",engine='interactive',shading=NA)

> head (Groceries) 
transacti ons in sparse format with 
6 transacti ons (rows) and 
169 items (columns) 
> inspect (head(Grocer i es)) 
n tems 
{citrus fruit, 
semi -finished bread, 
margarn ne, 
ready soups} 
{tropi cal fruit, 
yogurt , 
coff ee} 
{whole milk} 
{pip fruit, 
yogurt , 
cream cheese 
meat spreads} 
{other vegetables, 
whole milk, 
condensed milk, 
long life bakery product} 
{whole milk, 
butter , 
yogurt , 
r n ce, 
abrasive cleaner} 

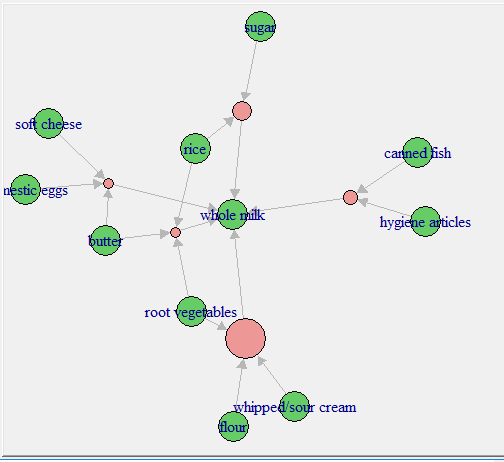
> summary(Groceries) 
transacti ons as itemMatrix in sparse format with 
9835 rows (elements/itemsets/tr ansacti ons) and 
169 columns (items) and a density of O. 02609146 
most frequent items: 
whole milk other vegetabl es 
2513 
soda 
1715 
1903 
yogurt 
1372 
rolls/buns 
1809 
(other) 
34055 
el ement (itemset/transaction) length distribution: 
s 1 zes 
2159 1643 1299 1005 855 
645 
14 
545 
20 
438 
21 
350 246 
22 
Max. 
32. 000 
182 
24 
117 
26 
14 15 
Min. 1st Qu. 
1. 000 
2. 000 
14 
Medi an 
3. 000 
Mean 3rd Qu. 
4. 409 
6. 000 
i ncl udes extended item information 
exampl es : 
labels level 2 
levell 
1 frankfurter sausage meat and sausage 
sausage sausage meat and sausage 
3 1 iver loaf sausage meat and sausage 

009 乙 
000 乙 
009 し 
000 し 
009 
(alnlosqe) fouanba 」 4山a引 

> rules—apriori (Grocer ies,parameter 
Apri ori 
Par ameter specification: 
list (supp—O. 001 , conf—O. 8)) 
confidence minval smax arem aval original support maxtime 
1 none FALSE 
support minlen max len target ext 
10 rules TRUE 
o. 001 
TRUE 
Al gorithmic control : 
fi Iter tree heap memopt 
0.1 TRUE TRUE FALSE 
Absolute minimum support 
load sort verbose 
TRUE 
count: g 
TRUE 
set item appearances 
. [O item(s)) done CO. ODs J. 
. [169 item(s), 9835 transaction(s)) done CO. Ols) . 
set transactions . 
sorting and recoding items ... 
[157 item(s)] done CO. ODs). 
creating transaction tree . 
done CO. Ols]. 
checking subsets of size 1 2 3 4 5 6 done CO. 06sJ. 
wr iti ng 
(410 rule(s)) done CO. ODs]. 
creating s4 object 
. done CO. Ols). 

> inspect (head(rules)) 
lhs 
{liquor , 
red/blush wine} 
{bottled beer} 
s upport 
o. 001931876 
o. 001016777 
o. 001728521 
o. 001016777 
o. 001118454 
o. 001321810 
confidence 
o. 9047619 
O. gogogog 
o. 8095238 
o. 8333333 
o. 9166667 
o. 8125000 
cover age 
lift count 
o. 002135231 11. 235269 
{curd , 
cereals} 
{yogurt , 
cereals} 
{butter , 
{soups , 
bottled beer} 
{napki ns , 
house keeping 
pr oducts} 
{whol e 
{whol e 
{whol e 
{whol e 
{whol e 
mi 1k} 
mi 1k} 
mi 1k} 
mi 1k} 
mi 1k} 
o. 0011184 54 
o. 002135231 
o. 001220132 
o. 001220132 
o. 00162684 3 
3. 557863 
3. 168192 
3. 261374 
3. 587512 
3. 179840 

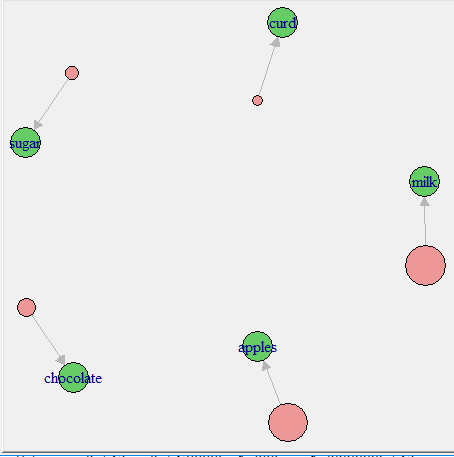
> inspect (rules [1:10)) 
lhs 
{liquor ,red/blush wine} 
{curd , cereals} 
{yogurt , cereal s} 
{butter , jam} 
{soups , bottled beer} 
{napki ns , house keeping products} 
{whi pped/sour cream , house keepi ng 
{pastry , sweet spr eads} 
{turkey , curd} 
{ri ce,sugar} 
{bottled beer} 
products} 
{who 1 e 
{who 1 e 
{who 1 e 
{who 1 e 
{who 1 e 
{who 1 e 
{who 1 e 
{other 
{who 1 e 
mi 1k} 
mi 1k} 
mi 1k} 
mi 1k} 
mi 1k} 
mi 1k} 
vegetabl es} 
support 
o. 001931876 
o. 001016777 
o. 001728521 
o. 001016777 
o. 0011184 54 
o. 001321810 
o. 001220132 
o. 001016777 
o. 001220132 
o. 001220132 
conf i dence 
o. 9047619 
O. gogogog 
o. 8095238 
o. 8333333 
o. 9166667 
o. 8125000 
o. 9230769 
O. gogogog 
o. 8000000 
1. 0000000 
cover age 
o. 002135231 
o. 001118454 
o. 002135231 
o. 001220132 
o. 001220132 
o. 001626843 
o. 001321810 
o. 001118454 
o. 001525165 
o. 001220132 
lift 
11. 235269 
3. 557863 
3.168192 
3. 261374 
3. 587512 
3.179840 
3. 612599 
3. 557863 
4.134524 
3. 913649 
count 
19 
10 
17 
10 
11 
13 
12 
10 
12 
12 



> *will not work 
> bakeryrules=apriori (bakery , parameter 
list (supp—O. 001 , conf—O. 8)) 
Apr i ori 
Parameter specification: 
confidence minval smax 
1 
tar get ext 
rules TRUE 
Al gorithmic control : 
fi Iter tree heap memopt 
0.1 TRUE TRUE FALSE 
Absolute minimum support 
set item appearances 
ar em 
aval originalsupport maxtime support minlen max len 
TRUE 
5 0. 001 
load sort verbose 
TRUE 
count: 3 
TRUE 
. [O item(s)) done CO. ODs J. 
. [g item(s), 3537 transaction(s)) done [O. . 
set transactions . 
sorting and recoding items . 
creating transaction tree . 
checking subsets of size 1 2 
writing . 
CO rule(s)) done 
creating s4 object 
. done 
[g item(s)] done CO. ODs). 
done CO. OOs]. 
3 done CO. OOs]. 
co. oos). 
co. oos). 

> i (head(bakeryTransacs)) 
Items 
{appl es , curd, sugar} 
{milk, sugar} 
{curd , mi 1k} 
{apples , curd, sugar} 
{appl es , chocolate , sugar 

> bakeryrules=apriori (bakeryTransacs ,parameter 
Apri ori 
Par ameter specification: 
1 i st (supp—O. 002 , conf—O. 08)) 
confidence minval smax arem aval originalsupport maxtime support minlen max len 
o. 08 
target ext 
rules TRUE 
Algorithmic control : 
1 none FALSE 
TRUE 
5 0. 002 
fi Iter tree heap memopt load sort verbose 
0.1 TRUE TRUE FALSE TRUE 
Absolute minimum support count: 2 
TRUE 
set item appearances 
. [O item(s)) done CO. ODs J. 
set transactions . 
. [5 item(s), 1000 transaction(s)) done 
sorting and recoding items ... 
item(s)] done CO. ODs). 
creating transaction tree . 
done CO. OOs]. 
checking subsets of size 1 2 3 4 5 done CO. ODs J. 
writing . 
[80 rule(s)) done CO. ODs]. 
creating s4 object 
. done CO.OOs). 
> inspect (head(bakeryrules)) 
lhs 
{curd} 
rhs 
{curd} 
{chocol ate} 
{sugar} 
{milk} 
{apples} 
{choco 1 ate} 
support confi dence 
0. 493 
o. 506 
0. 498 
o. 542 
o. 540 
o. 231 
o. 4930000 
o. 5060000 
o. 4980000 
o. 5420000 
o. 5400000 
o. 4685598 
cover age 
1. 000 
1. 000 
1. 000 
1. 000 
1. 000 
0.493 
lift 
1. 0000000 
1. 0000000 
1. 0000000 
1. 0000000 
1. 0000000 
o. 9260076 
count 
493 
506 
498 
542 
540 
231 



# 

# QUESTION 5

**Program 5 : Use Naive bayes, K-nearest, and Decision tree classification algorithms and build classifiers. Divide the data set into training and test sets. Compare the accuracy of the different classifiers under the following situations:**

**5.1 a) Training set = 75% Test set = 25%**

**b) Training set = 66.6% (2/3rd of total), Test set = 33.3%**

**5.2 Training set is chosen by i)hold out method ii)Random subsampling iii)Cross-Validation. Compare the accuracy of the classifiers obtained.**

**5.3 Data is scaled to standard format.**

**IRIS Dataset**

#install.packages("caret",dependencies = TRUE)

library(caret)

#data scaled to standard format------------------

preprocessValue=preProcess(iris[,],method = c("scale","center"))

iris\_trans=predict(preprocessValue,iris[,])

sapply(iris\_trans[,-5],sd)

summary(iris\_trans)

#using sample

#iris data set

#1)r-part==========================

#hold out method(75%)--------------------------

n=nrow(iris\_trans)

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

model=train(Species~.,data=iris\_train,'rpart')

pred=predict(model,iris\_test[,-5])

conf=confusionMatrix(pred,iris\_test$Species)$table

dt\_acchold75=((sum(diag(conf)))/sum(conf))\*100

#hold out method(66%)--------------------------

n=nrow(iris\_trans)

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

model=train(Species~.,data=iris\_train,'rpart')

pred=predict(model,iris\_test[,-5])

conf=confusionMatrix(pred,iris\_test$Species)$table

dt\_acchold66=((sum(diag(conf)))/sum(conf))\*100

#random sub sampling(75%)-------------------------

n=nrow(iris\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

model=train(Species~.,data=iris\_train,'rpart')

pred=predict(model,iris\_test[,-5])

conf=confusionMatrix(pred,iris\_test$Species)$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

dt\_accsubsample75=acctotal/k

cat("dt Aggregate(75%) :",dt\_accsubsample75)

#random sub sampling(66%)-------------------------

n=nrow(iris\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

model=train(Species~.,data=iris\_train,'rpart')

pred=predict(model,iris\_test[,-5])

conf=confusionMatrix(pred,iris\_test$Species)$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

dt\_accsubsample66=acctotal/k

cat("dt Aggregate(66%) :",dt\_accsubsample66)

#cross validation-------------------------

model=train(iris\_trans[,-5],iris\_trans$Species,'rpart',trControl=trainControl(method = 'cv',number = 10))

pred=predict(model,iris\_trans[,-5])

conf=confusionMatrix(pred,iris\_trans$Species)$table

dt\_acccv=((sum(diag(conf)))/sum(conf))\*100

#comparing all accuracies----------------

barplot(c(dt\_acchold75,dt\_acchold66,dt\_accsubsample75,dt\_accsubsample66,dt\_acccv),xlab="decision tree method",ylab="accuracies(in %)"

,ylim = c(0,100), names.arg=c("dt\_acchold75","dt\_acchold66","dt\_accsubsample75","dt\_accsubsample66","dt\_acccv"),col="red")

#install.packages("e1071",dependencies = TRUE)

library(e1071)

#2)naive-Bayes classifier=====================

#hold out method(75%)--------------------------

n=nrow(iris\_trans)

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

model=naiveBayes(Species~.,data=iris\_train)

pred=predict(model,iris\_test[,-5])

conf=confusionMatrix(pred,iris\_test$Species)$table

nb\_acchold75=((sum(diag(conf)))/sum(conf))\*100

#hold out method(66%)--------------------------

n=nrow(iris\_trans)

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

model=naiveBayes(Species~.,data=iris\_train)

pred=predict(model,iris\_test[,-5])

conf=confusionMatrix(pred,iris\_test$Species)$table

nb\_acchold66=((sum(diag(conf)))/sum(conf))\*100

#random sub sampling(75%)-------------------------

n=nrow(iris\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

model=naiveBayes(Species~.,data=iris\_train)

pred=predict(model,iris\_test[,-5])

conf=confusionMatrix(pred,iris\_test$Species)$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

nb\_accsubsample75=acctotal/k

cat("nb Aggregate(75%) :",nb\_accsubsample75)

#random sub sampling(66%)-------------------------

n=nrow(iris\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

model=naiveBayes(Species~.,data=iris\_train)

pred=predict(model,iris\_test[,-5])

conf=confusionMatrix(pred,iris\_test$Species)$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

nb\_accsubsample66=acctotal/k

cat("nb Aggregate(66%) :",nb\_accsubsample66)

#cross validation-------------------------

model=naiveBayes(iris\_trans[,-5],iris\_trans$Species,trControl=trainControl(method = 'cv',number = 10))

pred=predict(model,iris\_trans[,-5])

conf=confusionMatrix(pred,iris\_trans$Species)$table

nb\_acccv=((sum(diag(conf)))/sum(conf))\*100

#comparing all accuracies----------------

barplot(c(nb\_acchold75,nb\_acchold66,nb\_accsubsample75,nb\_accsubsample66,nb\_acccv),xlab="naive-Bayes method",ylab="accuracies(in %)"

,ylim = c(0,100), names.arg=c("nb\_acchold75","nb\_acchold66","nb\_accsubsample75","nb\_accsubsample66","nb\_acccv"),col="red")

#3)knn ==========================

library(class)

#hold out method(75%)--------------------------

n=nrow(iris\_trans)

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

classlabels=iris\_train[,5]

testclasslabels=iris\_test[,5]

model=knn(train = iris\_train[,-5],test = iris\_test[,-5],cl=classlabels,k=13)

conf=confusionMatrix(testclasslabels,model)$table

knn\_acchold75=((sum(diag(conf)))/sum(conf))\*100

#hold out method(66%)--------------------------

n=nrow(iris\_trans)

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

classlabels=iris\_train[,5]

testclasslabels=iris\_test[,5]

model=knn(train = iris\_train[,-5],test = iris\_test[,-5],cl=classlabels,k=13)

conf=confusionMatrix(testclasslabels,model)$table

knn\_acchold66=((sum(diag(conf)))/sum(conf))\*100

#random sub sampling(75%)-------------------------

n=nrow(iris\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

classlabels=iris\_train[,5]

testclasslabels=iris\_test[,5]

model=knn(train = iris\_train[,-5],test = iris\_test[,-5],cl=classlabels,k=13)

conf=confusionMatrix(testclasslabels,model)$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

knn\_accsubsample75=acctotal/k

cat("knn Aggregate(75%) :",knn\_accsubsample75)

#random sub sampling(66%)-------------------------

n=nrow(iris\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.66\*n),replace = FALSE)

iris\_train=iris\_trans[trainIndex,]

iris\_test=iris\_trans[-trainIndex,]

classlabels=iris\_train[,5]

testclasslabels=iris\_test[,5]

model=knn(train = iris\_train[,-5],test = iris\_test[,-5],cl=classlabels,k=13)

conf=confusionMatrix(testclasslabels,model)$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

knn\_accsubsample66=acctotal/k

cat("knn Aggregate (66%):",knn\_accsubsample66)

#cross validation-------------------------

classlabels=iris\_trans[,5]

model=knn.cv(train = iris\_trans[,-5],cl=classlabels,k=13)

conf=confusionMatrix(iris\_trans$Species,model)$table

knn\_acccv=((sum(diag(conf)))/sum(conf))\*100

#comparing all accuracies----------------

barplot(c(knn\_acchold75,knn\_acchold66,knn\_accsubsample75,knn\_accsubsample66,knn\_acccv),xlab="knn method",ylab="accuracies(in %)"

,ylim = c(0,100), names.arg=c("knn\_acchold75","knn\_acchold66","knn\_accsubsample75","knn\_accsubsample66","knn\_acccv"),col="red")

#matrix of comparison of all models

chart=matrix(c(

c(dt\_acchold75,dt\_acchold66,dt\_accsubsample75,dt\_accsubsample66,dt\_acccv),

c(nb\_acchold75,nb\_acchold66,nb\_accsubsample75,nb\_accsubsample66,nb\_acccv),

c(knn\_acchold75,knn\_acchold66,knn\_accsubsample75,knn\_accsubsample66,knn\_acccv)

),5,3,

dimnames = list(

c("accuracy\_holdout\_75%","accuracy\_holdout\_66%","accuracy\_subsample\_75%","accuracy\_subsample\_66%","accuracy\_cross\_vaidation"),

c("decision tree","naive bayes","KNN")

)

)

#using caTool

library(caTools)

set.seed(123)

split=sample.split(iris$Species,SplitRatio = 0.75)

training\_set=subset(iris,split==TRUE)

test\_set=subset(iris,split==FALSE)

dim(training\_set)

dim(test\_set)

x=training\_set

y=rownames(training\_set)

model=train(x,y,'rpart',trControl=trainControl(method = 'cv',number = 10))

#using random numbers by sample

s=sample(150,50)

iris\_train=iris[s,]

iris\_test=iris[-s,]

dim(iris\_train)

dim(iris\_test)

x=iris\_train

y=rownames(iris\_train)

model=train(x,y,'rpart',trControl=trainControl(method = 'cv',number = 10))

**Breast Cancer Dataset**

cancer=read.csv("C:\\Users\\Girish\\Desktop\\paras\\DM\\breast-cancer-wisconsin.csv")

#pre processing

cancer\_preprcs=cancer[,2:11]

sum(is.na(cancer\_preprcs))

sum(complete.cases(cancer\_preprcs))

#conversion of y class labels from {2,4} to {benign,malignant}

cancer\_preprcs$Class=cancer\_preprcs$Class/2

v=c("benign","malignant")

cancer\_preprcs$Class=v[cancer\_preprcs$Class]

#remove ? values in 6th column

for (i in 1:10) {

cat(i,which(cancer\_preprcs[,i]=="?"),"\n")

}

length(which(cancer\_preprcs[,6]=="?"))

#cancer\_preprcs$Bare\_Nuclei[which(cancer\_preprcs[,6]=="?")]=NA

cancer\_preprcs=cancer\_preprcs[-which(cancer\_preprcs[,6]=="?"),]

#important: type of bare nuclei is character convert to integer

typeof(cancer\_preprcs$Bare\_Nuclei)

cancer\_preprcs$Bare\_Nuclei=as.integer(cancer\_preprcs$Bare\_Nuclei)

typeof(cancer\_preprcs$Bare\_Nuclei)

# y values should be {factor}categorical(here after conversion{from int to categorical} they become character so convert them into factor)

typeof(cancer\_preprcs$Class)

cancer\_preprcs$Class=as.factor(cancer\_preprcs$Class)

typeof(cancer\_preprcs$Class)

str(cancer\_preprcs)

# no need of scaling because all values are between 1-10

#install.packages("caret",dependencies = TRUE)

library(caret)

#data scaled to standard format------------------

preprocessValue=preProcess(cancer\_preprcs[,],method = c("scale","center"))

cancer\_trans=predict(preprocessValue,cancer\_preprcs[,])

sapply(cancer\_trans[,-10],sd)

summary(cancer\_trans)

#cancer\_trans=cancer\_preprcs

#using sample

#breast cancer data set

#1)r-part==========================

#hold out method(75%)--------------------------

n=nrow(cancer\_trans)

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

model=train(Class~.,data=cancer\_train,'rpart')

pred=predict(model,cancer\_test[,-10])

conf=confusionMatrix(as.factor(pred),as.factor(cancer\_test$Class))$table

dt\_acchold75=((sum(diag(conf)))/sum(conf))\*100

#hold out method(66.6%)--------------------------

n=nrow(cancer\_trans)

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

model=train(Class~.,data=cancer\_train,'rpart')

pred=predict(model,cancer\_test[,-10])

conf=confusionMatrix(as.factor(pred),as.factor(cancer\_test$Class))$table

dt\_acchold66=((sum(diag(conf)))/sum(conf))\*100

#random sub sampling(75%)-------------------------

n=nrow(cancer\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

model=train(Class~.,data=cancer\_train,'rpart')

pred=predict(model,cancer\_test[,-10])

conf=confusionMatrix(as.factor(pred),as.factor(cancer\_test$Class))$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

dt\_accsubsample75=acctotal/k

cat("dt Aggregate(75%) :",dt\_accsubsample75)

#random sub sampling(66%)-------------------------

n=nrow(cancer\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

model=train(Class~.,data=cancer\_train,'rpart')

pred=predict(model,cancer\_test[,-10])

conf=confusionMatrix(as.factor(pred),as.factor(cancer\_test$Class))$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

dt\_accsubsample66=acctotal/k

cat("dt Aggregate(66%) :",dt\_accsubsample66)

#cross validation-------------------------

model=train(cancer\_trans[,-10],cancer\_trans$Class,'rpart',trControl=trainControl(method = 'cv',number = 10))

pred=predict(model,cancer\_trans[,-10])

conf=confusionMatrix(as.factor(pred),as.factor(cancer\_trans$Class))$table

dt\_acccv=((sum(diag(conf)))/sum(conf))\*100

#comparing all accuracies----------------

barplot(c(dt\_acchold75,dt\_acchold66,dt\_accsubsample75,dt\_accsubsample66,dt\_acccv),xlab="decision tree method",ylab="accuracies(in %)"

,ylim = c(0,100), names.arg=c("dt\_acchold75","dt\_acchold66","dt\_accsubsample75","dt\_accsubsample66","dt\_acccv"),col="red")

#install.packages("e1071",dependencies = TRUE)

library(e1071)

#2)naive-Bayes classifier=====================

#hold out method(75%)--------------------------

n=nrow(cancer\_trans)

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

model=naiveBayes(Class~.,data=cancer\_train)

pred=predict(model,cancer\_test[,-10])

conf=confusionMatrix(pred,cancer\_test$Class)$table

nb\_acchold75=((sum(diag(conf)))/sum(conf))\*100

#hold out method(66%)--------------------------

n=nrow(cancer\_trans)

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

model=naiveBayes(Class~.,data=cancer\_train)

pred=predict(model,cancer\_test[,-10])

conf=confusionMatrix(pred,cancer\_test$Class)$table

nb\_acchold66=((sum(diag(conf)))/sum(conf))\*100

#random sub sampling(75%)-------------------------

n=nrow(cancer\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

model=naiveBayes(Class~.,data=cancer\_train)

pred=predict(model,cancer\_test[,-10])

conf=confusionMatrix(pred,cancer\_test$Class)$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

nb\_accsubsample75=acctotal/k

cat("nb Aggregate(75%) :",nb\_accsubsample75)

#random sub sampling(66%)-------------------------

n=nrow(cancer\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

model=naiveBayes(Class~.,data=cancer\_train)

pred=predict(model,cancer\_test[,-10])

conf=confusionMatrix(pred,cancer\_test$Class)$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

nb\_accsubsample66=acctotal/k

cat("nb Aggregate(66%) :",nb\_accsubsample66)

#cross validation-------------------------

model=naiveBayes(cancer\_trans[,-10],cancer\_trans$Class,trControl=trainControl(method = 'cv',number = 10))

pred=predict(model,cancer\_trans[,-10])

conf=confusionMatrix(pred,cancer\_trans$Class)$table

nb\_acccv=((sum(diag(conf)))/sum(conf))\*100

#comparing all accuracies----------------

barplot(c(nb\_acchold75,nb\_acchold66,nb\_accsubsample75,nb\_accsubsample66,nb\_acccv),xlab="naive-Bayes method",ylab="accuracies(in %)"

,ylim = c(0,100), names.arg=c("nb\_acchold75","nb\_acchold66","nb\_accsubsample75","nb\_accsubsample66","nb\_acccv"),col="red")

#3)knn ==========================

library(class)

#hold out method(75%)----------------------------

n=nrow(cancer\_trans)

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

classlabels=cancer\_train[,10]

testclasslabels=cancer\_test[,10]

model=knn(train = cancer\_train[,-10],test = cancer\_test[,-10],cl=classlabels,k=13)

conf=confusionMatrix(testclasslabels,model)$table

knn\_acchold75=((sum(diag(conf)))/sum(conf))\*100

#hold out method(66%)----------------------------

n=nrow(cancer\_trans)

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

classlabels=cancer\_train[,10]

testclasslabels=cancer\_test[,10]

model=knn(train = cancer\_train[,-10],test = cancer\_test[,-10],cl=classlabels,k=13)

conf=confusionMatrix(testclasslabels,model)$table

knn\_acchold66=((sum(diag(conf)))/sum(conf))\*100

#random sub sampling(75%)-------------------------

n=nrow(cancer\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.75\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

classlabels=cancer\_train[,10]

testclasslabels=cancer\_test[,10]

model=knn(train = cancer\_train[,-10],test = cancer\_test[,-10],cl=classlabels,k=13)

conf=confusionMatrix(testclasslabels,model)$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

knn\_accsubsample75=acctotal/k

cat("knn Aggregate(75%) :",knn\_accsubsample75)

#random sub sampling(66%)-------------------------

n=nrow(cancer\_trans)

k=10

i=0

acctotal=0

while(i<k){

trainIndex=sample(1:n,size=round(0.666\*n),replace = FALSE)

cancer\_train=cancer\_trans[trainIndex,]

cancer\_test=cancer\_trans[-trainIndex,]

classlabels=cancer\_train[,10]

testclasslabels=cancer\_test[,10]

model=knn(train = cancer\_train[,-10],test = cancer\_test[,-10],cl=classlabels,k=13)

conf=confusionMatrix(testclasslabels,model)$table

acc=((sum(diag(conf)))/sum(conf))\*100

acctotal=acctotal+acc

print(acc)

i=i+1

}

knn\_accsubsample66=acctotal/k

cat("knn Aggregate(66%) :",knn\_accsubsample66)

#cross validation-------------------------

classlabels=cancer\_trans[,10]

model=knn.cv(train = cancer\_trans[,-10],cl=classlabels,k=13)

conf=confusionMatrix(cancer\_trans$Class,model)$table

knn\_acccv=((sum(diag(conf)))/sum(conf))\*100

#comparing all accuracies----------------

barplot(c(knn\_acchold75,knn\_acchold66,knn\_accsubsample75,knn\_accsubsample66,knn\_acccv),xlab="knn method",ylab="accuracies(in %)"

,ylim = c(0,100), names.arg=c("knn\_acchold75","knn\_acchold66","knn\_accsubsample75","knn\_accsubsample66","knn\_acccv"),col="red")

#matrix of comparison of all models

chart=matrix(c(

c(dt\_acchold75,dt\_acchold66,dt\_accsubsample75,dt\_accsubsample66,dt\_acccv),

c(nb\_acchold75,nb\_acchold66,nb\_accsubsample75,nb\_accsubsample66,nb\_acccv),

c(knn\_acchold75,knn\_acchold66,knn\_accsubsample75,knn\_accsubsample66,knn\_acccv)

),5,3,

dimnames = list(

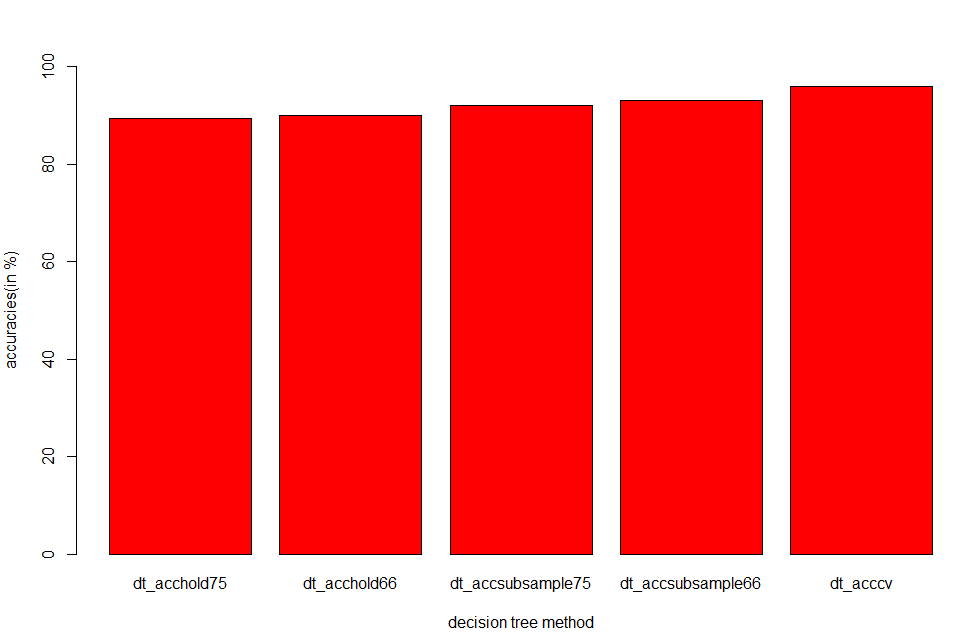
c("accuracy\_holdout\_75%","accuracy\_holdout\_66%","accuracy\_subsample\_75%","accuracy\_subsample\_66%","accuracy\_cross\_vaidation"),

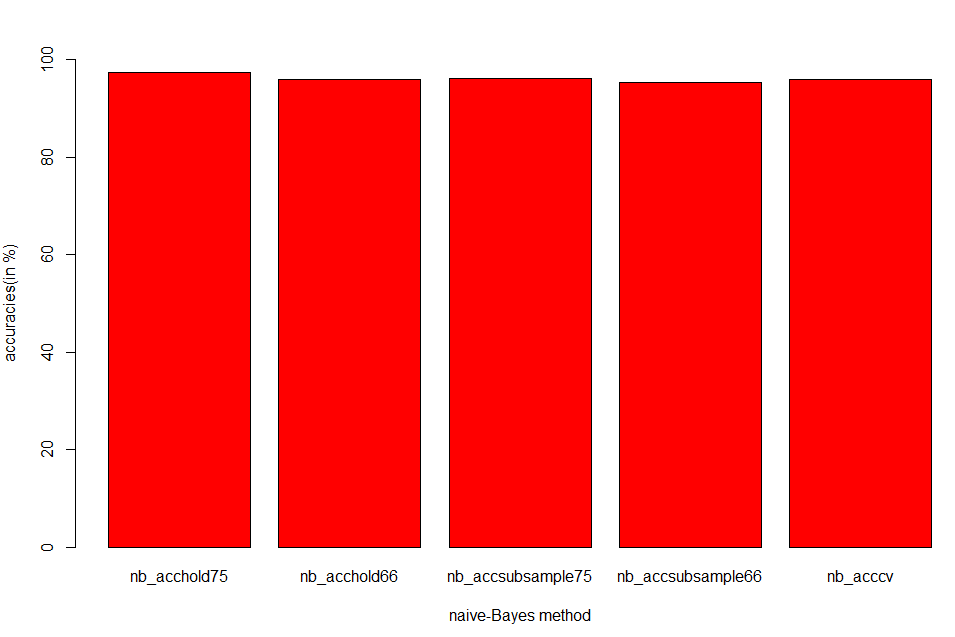
c("decision tree","naive bayes","KNN")

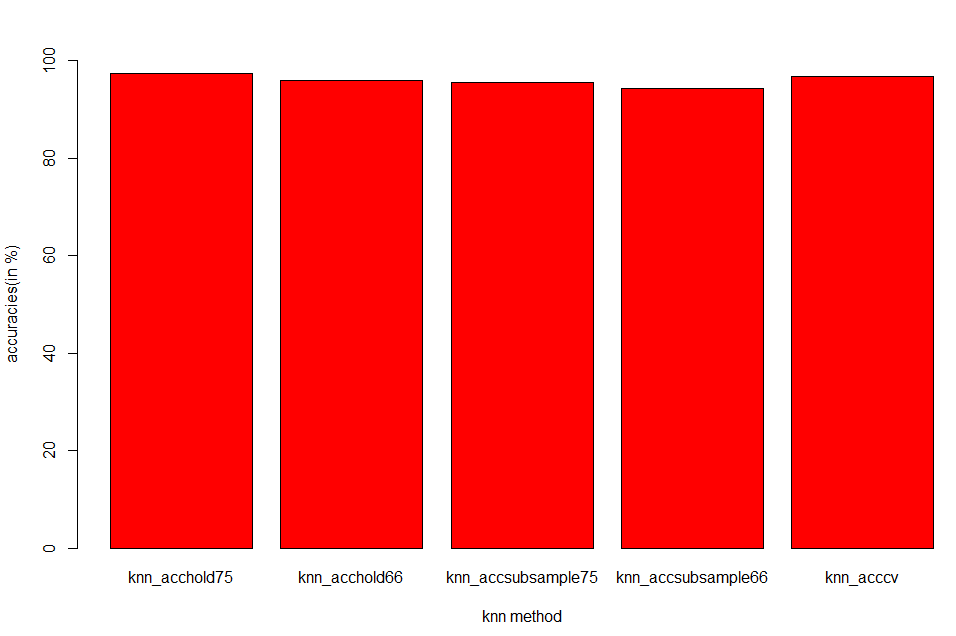
)

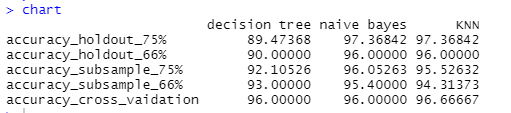
)

**IRIS SCREENSHOTS**

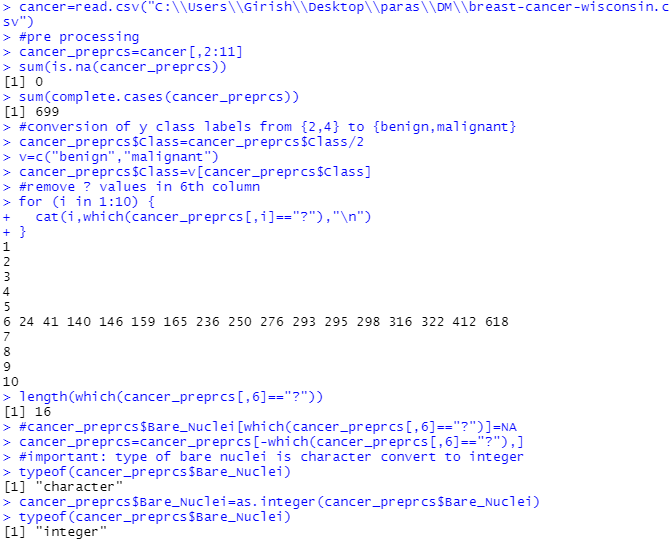


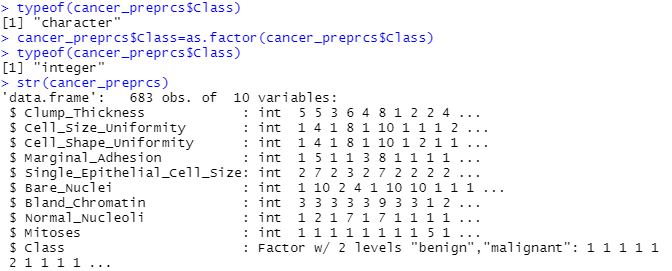


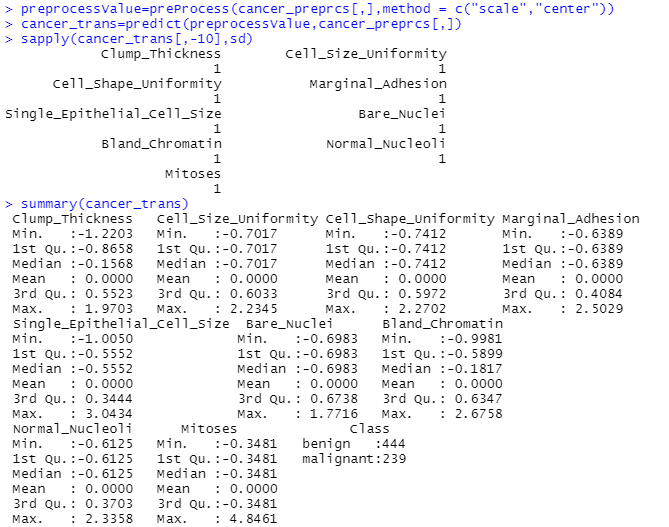


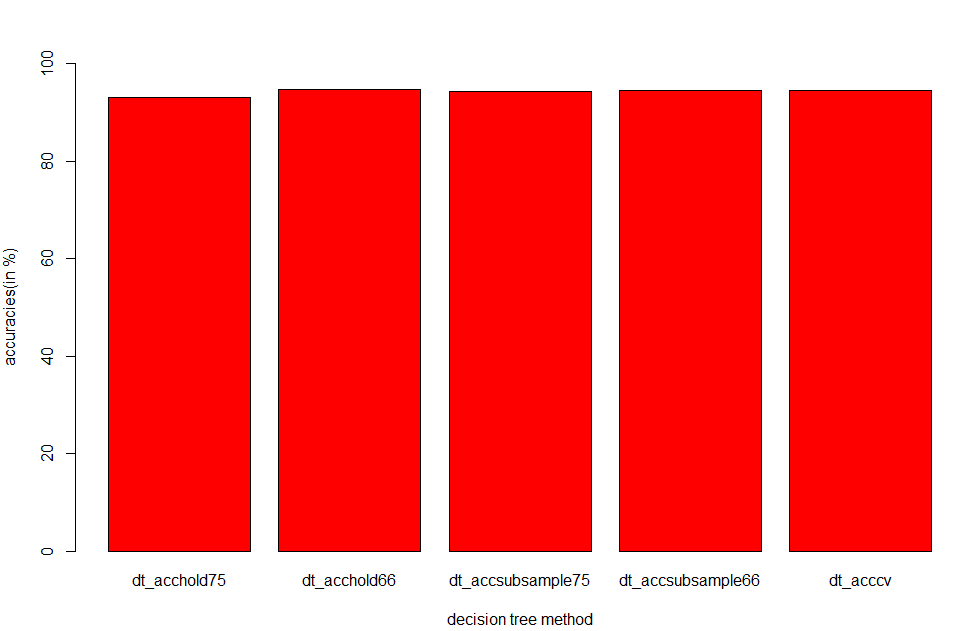


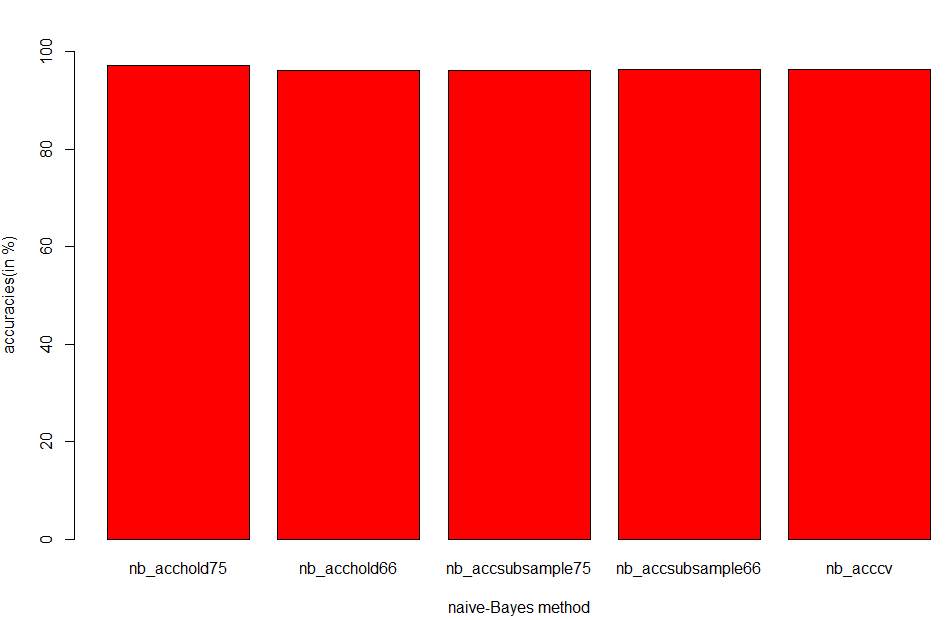
**breast-cancer-wisconsin SCREENSHOTS**

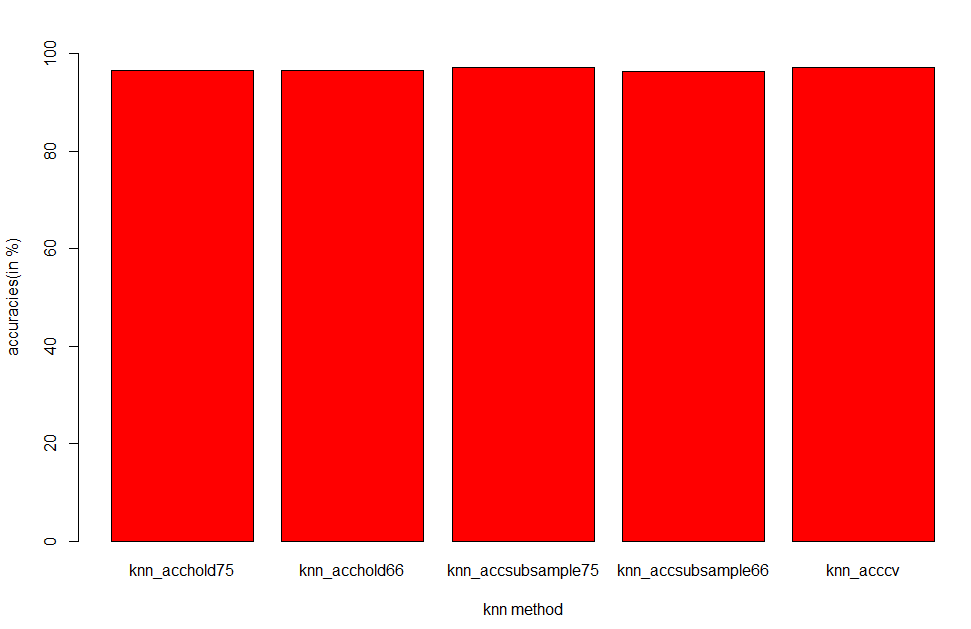


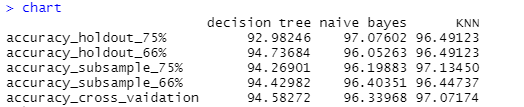












# 

# 

# QUESTION 6

**Program 6 : Use Simple K Means, DBScan, Hierarchical clustering algorithms for clustering. Compare the performance of clusters by changing the parameters involved in the algorithms.**

###--------------Iris data set --------------------------

#install.packages("caret",dependencies = TRUE)

library(caret)

library(ggplot2)

#data scaled to standard format------------------

preprocessValue=preProcess(iris[,],method = c("scale","center"))

iris\_trans=predict(preprocessValue,iris[,])

sapply(iris\_trans[,-5],sd)

summary(iris\_trans)

#### kmeans

k=list()

for(i in 1:10){

k[[i]]=kmeans(iris\_trans[,-5],i)

}

betweenss\_totss=list()

for(i in 1:10){

betweenss\_totss[[i]]=k[[i]]$betweenss/k[[i]]$totss

}

plot(1:10,betweenss\_totss,type = "b",ylab = "betweenss ss/total ss",xlab = "Cluster(k)")

for(i in 1:10){

print(table(iris\_trans$Species,k[[i]]$cluster))

}

ggplot(iris\_trans,aes(Petal.Length,Petal.Width,color=Species)) + geom\_point()

ggplot(iris\_trans,aes(Petal.Length,Petal.Width,color=k[[3]]$cluster)) + geom\_point()

#### hierarchical clustering

d=dist(iris\_trans)

fitH=hclust(d,"ward.D2")

plot(fitH)

rect.hclust(fitH,k=3,border = "red")

cluster<-cutree(fitH,3)

cluster

plot(iris\_trans,col=cluster)

table(iris\_trans$Species,cluster)

ggplot(iris\_trans, aes(Petal.Length, Petal.Width, color = Species)) +

geom\_point(alpha = 0.4, size = 3.5) + geom\_point(col = cluster) +

scale\_color\_manual(values = c('black', 'red', 'green'))

#### dB-Scan algorithm

#install.packages("dbscan")

library(dbscan)

kNNdistplot(iris\_trans[,-5],k=3) # to decide value of eps

abline(h=0.7,col="red",lty=2)

fitD = dbscan(iris\_trans[,-5],eps =0.7 ,minPts =5 )

plot(iris\_trans,col=fitD$cluster)

table(iris\_trans$Species,fitD$cluster)

###--------------HTRU\_2 data set --------------------------

#install.packages("caret",dependencies = TRUE)

library(caret)

library(ggplot2)

htru<-read.csv("C:\\Users\\Girish\\Desktop\\paras\\DM\\HTRU2\\HTRU\_2.csv",header=FALSE)

names(htru)<-c("Profile\_mean","Profile\_stdev","Profile\_skewness","Profile\_kurtosis"," DM\_mean","DM\_stdev","DM\_skewness", "DM\_kurtosis","class")

sum(is.na(htru)) #no missing values

sum(complete.cases(htru)) #all are complete cases

unique(htru[,9]) #two classes positive(1) and negative(0)

sum(htru[,9]) #number of positive classes(1639)

nrow(htru) - sum(htru[,9]) #number of negative classes(16259)

#data scaled to standard format------------------

preprocessValue=preProcess(htru[,],method = c("scale","center"))

htru\_trans=predict(preprocessValue,htru[,])

sapply(htru\_trans[,-9],sd)

summary(htru\_trans)

short\_htru=htru\_trans[1:(nrow(htru\_trans)/2),]

#### kmeans

k=list()

for(i in 1:10){

k[[i]]=kmeans(htru\_trans[,-9],i)

}

betweenss\_totss=list()

for(i in 1:10){

betweenss\_totss[[i]]=k[[i]]$betweenss/k[[i]]$totss

}

plot(1:10,betweenss\_totss,type = "b",ylab = "betweenss ss/total ss",xlab = "Cluster(k)")

for(i in 1:10){

print(table(htru\_trans$class,k[[i]]$cluster))

}

ggplot(htru\_trans,aes(Profile\_mean,Profile\_stdev,color=class)) + geom\_point()

ggplot(htru\_trans,aes(Profile\_mean,Profile\_stdev,color=k[[2]]$cluster)) + geom\_point()

#### hierarchical clustering

d=dist(short\_htru)

fitH=hclust(d,"ward.D2")

plot(fitH)

rect.hclust(fitH,k=4,border = "red")

cluster<-cutree(fitH,4)

Cluster

plot(short\_htru,col=cluster)

table(short\_htru$class,cluster)

###ggplot not working for hierarchical

ggplot(short\_htru, aes(Profile\_mean, Profile\_stdev, color = class)) +

geom\_point(alpha = 0.4, size = 3.5) + geom\_point(col = cluster) +

scale\_color\_manual(values = c('black', 'red', 'green'))

#### dB-Scan algorithm

#install.packages("dbscan")

library(dbscan)

kNNdistplot(htru\_trans[,-9],k=3) # to decide value of eps

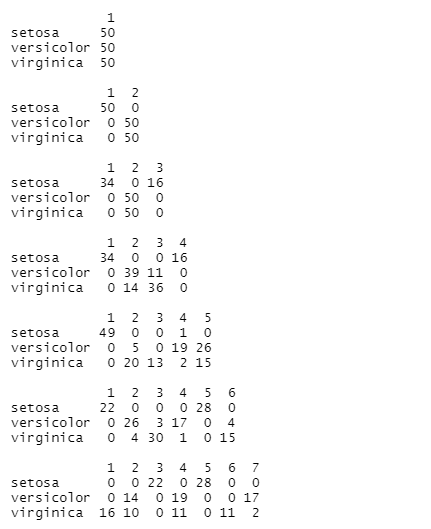
abline(h=0.7,col="red",lty=2)

fitD = dbscan(htru\_trans[,-9],eps =0.7 ,minPts =5 )

plot(htru\_trans,col=fitD$cluster)

table(htru\_trans$class,fitD$cluster)

02 
betweenss ss/total ss 
0.6 



setosa 
versi col or 
v n rgnm ca 
setosa 
versi color 
v n rgnm ca 
setosa 
versi color 
v n rgn m ca 
1 
o 
18 
o 
1 
21 
o 
1 
6 
o 
28 
o 
o 
o 
4 
15 
o 
4 
14 
4 
o 
o 
13 
15 
1 
8 
4 
o 
o 
3 24 
18 
4 
o 
17 
15 
o 
9 
o 
O 
16 
o 
15 
2 
O 
o 
26 
6 
O 
4 
14 
6 
16 
O 
o 
6 
o 
13 
1 
o 
O 
11 
o 
12 
20 
O 
O 
22 
O 
1 12 
4 
o 
4 
o 
o 
o 
9 
o 
o 
10 
18 
o 

Species 
setosa 
versicolor 
virginica 
Petal _ Length 

30 
25 
20 
1.5 
Petal_ Length 

Cluster Dendrogram 
d 
hclust C, 'Ward_D2") 

Cluster Dendrogram 
d 
hclust C, 'Ward_D2") 

clusterz-cutree(fitH, 3) 
Cluster 
[1] 
[46] 
[91] 
[1351 
1 
1 
1 
1 
1 
1 
1 
1 
1 
1 
1 
1 
1 
1 
2 
ı 
ı 
ı 
1 
322323 
333233 
2322 223333333332 
3333333333 2 33333 
2223233222 
3333333333 

10 
Sepal Length 
Sepal Width 
Petal Length 
Peta Width 
_os 
Spe cies 

cluster 
setosa 
versi col or 
v 1 rgn m ca 
1 
49 
o 
1 
27 
23 
2 48 

Petal WIdth 
Petal-Length 
0000330 
000 0 
Specles 
0 setosa 
VI rgl n - Ca 
versiCOlor 

100 
Points (sample) sorted by distance 
150 

10 
Sepal Length 
Sepal Width 
Petal Length 
Peta Width 
_os 
Spe cies 

> uster) 
setosa 
versi color 
v n rgn m ca 
2 48 
2 
50 

**HTRU-2 SCREENSHOTS**

#no missing values 
> sum(complete. cases (htru)) *all are complete cases 
Cll 17898 
> #two classes positive(l) and negative(O) 
> *number of positive classes (1639) 
Cll 1639 
> nrow(htru) 
g)) *number of negative cl asses (16259) 
Cll 16259 

pr eprocessvalue=pr eprocess (htru , ) , method 
c("scale" , "center'")) 
> 
(pr eprocessval ue , htru , l) 
-9) ,sd) 
> 
Pr of i I e_mean 
profi le_stdev profile_skewness profile_kurtosis 
DM_skewness 
summary (htru_trans) 
Pr of i I e_mean 
Min. 
:-4. 1035 
1st Qu. : 
-o. 3957 
Median . 
• o. 1559 
: o. 0000 
Mean 
3rd Qu. : 
o. 6239 
3.1785 
Min. 
:-o. 9736 
1st Qu. : 
-o. 6106 
Median . 
4040 
: o. 0000 
Mean 
3rd Qu. : 
o. 1079 
: 4. 3304 
Max. 
profi e_stdev 
Min. 
-3.18236 
1st Qu. : 
-o. 60988 
Median : 
o. 05815 
: o. 00000 
Mean 
3rd Qu. : 
o. 65374 
. 7. 63232 
DM_skewness 
Min. 
2. 53941 
1st Qu. : 
-o. 55970 
Median : 
o. 02884 
: o. 00000 
Mean 
3rd Qu. : 
o. 53248 
5. 82240 
Pr ofi 1 e_skewness 
Min. 
.-2. 212200 
1st Qu. : 
-0.423630 
Median : 
-o. 239293 
: o. 000000 
Mean 
3rd Qu. : 
-o. 004259 
: 7.134757 
Max. 
Min. 
:-1. 0030 
1st Qu. : 
-o. 6562 
Median : 
-o. 2046 
: o. 0000 
Mean 
3rd Qu. : 
o. 3234 
:10. 1971 
Max. 
profi e_kurtosis 
Min. 
1st Qu. 
Medi an 
Mean 
3rd Qu. 
Max. 
cl ass 
Min. 
1st Qu. 
Medi an 
Mean 
3rd Qu. 
-o. 5775 
-o. 3176 
-o. 2548 
o. 0000 
-o. 1366 
Min. 
: -o. 4208 
1st Qu. : 
-o. 3628 
Medi an 
:-o. 3329 
: o. 0000 
Mean 
3rd Qu. : 
-o. 2426 
: 7.1516 
-o. 3175 
-o. 3175 
-o. 3175 
o. 0000 
-o. 3175 
3. 1495 

bebveenss ss/total ss 
02 
0.6 

-o. 317490276032223 
3. 14952678340934 
-o. 317490276032223 
3. 14952678340934 
-o. 317490276032223 
3. 14952678340934 
-o. 317490276032223 
3. 14952678340934 
-o. 317490276032223 
3. 14952678340934 
-o. 317490276032223 
3. 14952678340934 
-o. 317490276032223 
3. 14952678340934 
-o. 317490276032223 
3. 14952678340934 
-o. 317490276032223 
3. 14952678340934 
16259 
1639 
776 
1281 
15 
972 
3062 
2370 
24 
1261 
16 
692 
109 
733 
858 
147 
15483 
358 
1393 
315 
16 
962 
14851 
352 
12075 
408 
1106 
233 
1040 
212 
4350 
113 
6576 
644 g 
29 
452 
8682 
73 
3295 
443 
3654 
669 
26 
675 
10 
4157 
874 
456 
10 
6473 
865 
33 
704 
1392 
8 
246 
2 2798 
490 
24 
374 
592 
68 
870 
169 
10 
869 
2024 
152 
3515 
202 
3231 
339 
482 
6 
5032 
3732 
186 
1985 
3876 

-Û. 317490276032223 
3. 14952678340934 
2924 
17 
919 
155 
4151 
606 
330 
1671 
3118 
77 
675 
9 
ı 
452 
10 
2512 
152 

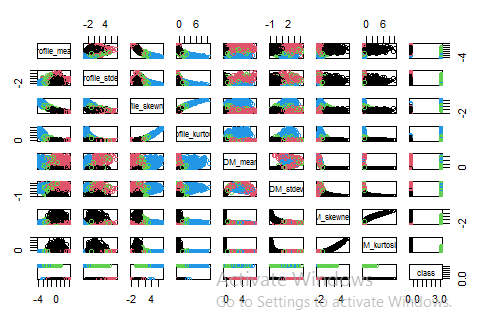
class 
Profile mean 

Proflle stde\/ 
k 冖 冖 2]]Scluste 乛 

Cluster Dendrogram 
d 
hclust C, 'Ward_D2") 

Cluster Dendrogram 
d 
hclust C, 'Ward_D2") 

> 
> 
cl CH , 4) 
cl uster 
20 
39 
115 
134 
153 
172 
191 
210 
229 
21 
40 
116 
135 
154 
173 
192 
211 
230 
22 
41 
117 
136 
155 
174 
193 
212 
231 
42 
99 
118 
137 
156 
175 
194 
213 
232 
24 
43 
100 
119 
138 
157 
176 
195 
214 
233 
25 
44 
63 
101 
120 
139 
158 
177 
196 
215 
234 
64 
83 
102 
121 
140 
159 
178 
197 
216 
235 
65 
84 
103 
122 
141 
160 
179 
198 
217 
236 
85 
104 
123 
142 
161 
180 
199 
218 
237 
105 
124 
143 
162 
181 
200 
219 
238 
106 
125 
144 
163 
182 
201 
220 
239 
107 
126 
145 
164 
183 
202 
221 
240 
13 
70 
108 
127 
146 
165 
184 
203 
222 
241 
14 
71 
90 
109 
128 
147 
166 
185 
204 
223 
242 
15 
53 
72 
91 
110 
129 
148 
167 
186 
205 
224 
243 
54 
92 
111 
130 
14 g 
168 
187 
206 
225 
244 
55 
74 
93 
112 
131 
150 
169 
188 
207 
226 
245 
75 
94 
113 
132 
151 
170 
189 
208 
227 
246 
114 
133 
152 
171 
1 go 
209 
228 
247 

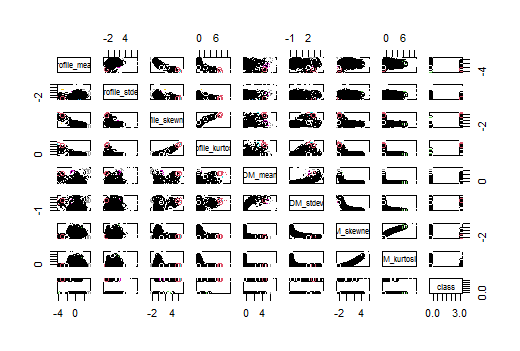


> , cluster) 
cluster 
-o. 317490276032223 6993 790 
3. 14952678340934 
o 
707 
456 

> aes (profi le_mean, profile_stdev , 
co or 
class)) + 
geom_poi nt (al pha 
0.4, size — 
3. 5) + geom_point (col 
cluster) + 
scal e_col or_manual (val ues 
c('black', 'red' 
' green')) 
Error: continuous value supplied to discrete scale 

Z 
10000 
Points (sample) sorted by distance 
15000 

Z 
10000 
Points (sample) sorted by distance 
15000 



-Û. 317490276032223 
3. 14952678340934 
-Û. 317490276032223 
3. 14952678340934 
333 
221 
15881 
1362 