R Textbook Companion for Data Mining: Concepts and Techniques by Jiawei Han, Micheline Kamber, and Jian Pei[1]

Created by
Shankru Guggari
Ph.D.
Computer Science and Engineering
BMS College of Engineering, Bengaluru
Cross-Checked by
R TBC Team

June 9, 2020

[1]Funded by a grant from the National Mission on Education through ICT - http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and R codes written in it can be downloaded from the "Textbook Companion Project" section at the website - https://r.fossee.in.

Book Description

Title: Data Mining: Concepts and Techniques

Author: Jiawei Han, Micheline Kamber, and Jian Pei

Publisher: Morgan Kaufmann, USA

Edition: 3

Year: 2011

ISBN: 9780123814791

R numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means an R code whose theory is explained in Section 2.3 of the book.

Contents

List of R Codes 4

2 Getting to know your Data 5

3 Data Preprocessing 12

4 Data warehousing and online analytical processing 16

5 Data Cube Technology 25

6 Mining frequent patterns associations and correlations basic concepts and methods 33

7 Advanced Pattern mining 34

8 Classification basic concepts 39

9 Classification Advanced methods 42

10 Cluster analysis basic concepts and mehtods 44

11 Advanced cluster analysis 49

12 Outlier detection 53

List of R Codes

Exa 2.6.1	Mean	5
Exa 2.7	Median	5
Exa 2.8	Mode	5
Exa 2.9	Midrange	6
Exa 2.10	Interquartile Range	6
Exa 2.11	Boxplot	7
Exa 2.12	Variance and standard deviation	7
Exa 2.13	Quantile Plot	7
Exa 2.15	Histogram	8
Exa 2.18	Dissimilarity between binary attributes	8
Exa 2.19	Euclidean distance and manhattan distance	9
Exa 2.20	Supremum distance	9
Exa 2.21	Dissimilarity between ordinal attributes	10
Exa 2.23	cosine similarity between two term frequency vectors	11
Exa 3.1	Correln analysis of nominal attributes using chi2	12
Exa 3.2	Covariance analysis of numeric attribute	12
Exa 3.3	Histograms	13
Exa 3.4	Min Max normalization	13
Exa 3.5	Z score normalization	14
Exa 3.6	Decimal Scalling	14
Exa 4.3	Fact Constellation	16
Exa 4.4	OLAP operations	18
Exa 4.6	A data cube is a lattice of cuboids	20
Exa 4.8	Join Index	22
Exa 4.9	OLAP query processing	23
Exa 5.9	Construct the inverted index	25
Exa 5.10	Compute shell fragment	28
Exa 5.11	Computing cubes with average measure	32

Exa 6.9 Correlation analysis using chi2 33
Exa 7.1 Mining multilevel association rules 34
Exa 7.2 redundancy among multilevel association rules 34
Exa 7.3 Rare patterns and negative patterns 35
Exa 7.6 Negatively related patterns 35
Exa 7.12 closed and maximal itemsets 36
Exa 7.13 pattern distance 37
Exa 7.15 Semantic annotations of a frequent patterns 37
Exa 8.9 Sensitivity and Specificity 39
Exa 8.10 Precision and recall 40
Exa 8.11 ROC Curve 41
Exa 9.1 Backpropagation algorithm 42
Exa 9.3 Error correcting codes 43
Exa 10.1 Clustering by K means partitioning 44
Exa 10.2 Drawback of k means 44
Exa 10.3 Agglomerative versus divisive hierachical clustering . . 46
Exa 10.4 Single versus complete linkages 47
Exa 10.7 Density reachability and density connectivity 47
Exa 10.8 core distance and reachability distance 48
Exa 11.5 Probabilistic clusters 49
Exa 11.7 Fuzzy clustering using the EM algorithm 49
Exa 11.14 clustering in a derived space 50
Exa 11.16 Bipartite graph 50
Exa 11.19 Measurements based on geodesic distance 51
Exa 11.21 cuts and clusters 51
Exa 11.23 Hard and soft constraints 52
Exa 12.1 Outliers 53
Exa 12.7 Detecting outliers using clustering 53
Exa 12.8 Univariate outliers detection using maxumum likelihood 54
Exa 12.9 Multivariate outlier detection using mahalanobis distance 54
Exa 12.10 Multivariate outlier detection using the chi2 statistic . 54
Exa 12.12 Multivariate outlier detection using multiple clusters . 55
Exa 12.13 Outlier detection using a histogram 55
Exa 12.14 local proximity based outliers 55
Exa 12.15 Detecting outliers as objects that do not belong to any clusters 56

Exa 12.16 clustering based outliers detection using distance to the closest cluster 56
Exa 12.18 detecting outliers in small clusters 57
Exa 12.20 Outlier detection by semi supervised learning 57
Exa 12.24 Outliers in subspace 58
Exa 12.25 angle based outliers 58

Chapter 2

Getting to know your Data

R code Exa 2.6.1 Mean

```
Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)

print("Mean")

print(paste("$",mean(Data)))
```

R code Exa 2.7 Median

```
Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)
print("Median")
print(paste("$",median(Data)))
```

R code Exa 2.8 Mode

```
Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)

```

```
mode <- function(x) {
   uni_value <- unique(x)
    uni_value[which.max(tabulate(match(x, uni_value)))]
}

print("Mode")
print(paste("$",mode(Data)))
```

R code Exa 2.9 Midrange

```
Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)


print("Mid range")


Mid_Range<- ((min(Data)+max(Data))/2)

pr_mir<- Mid_Range

print(paste("$",pr_mir))
```

R code Exa 2.10 Interquartile Range

```
Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)


print("Interquartile Range")


print(IQR(Data))
```

R code Exa 2.11 Boxplot

```
Data <- data.frame(MG= c(30,36,47,50,52), CY=c
    (25,60,30,21,70))


boxplot(Data,xlab = "Number of Cylinders",ylab = "
    Miles Per Gallon", main = "Summary of Mileage")
```

R code Exa 2.12 Variance and standard deviation

```
Data <- c(30,36,47,50,52,52,56,60,63,70,70,110)

print("variance")
print(var(Data))


print("Standard Deviation")

######"The  answer provided in the textbook is
    19.47"
print(sd(Data))
```

R code Exa 2.13 Quantile Plot

```
Unit_price = c(40,43,47,74,75,78,115,117,120)
count_of_items_sold =c
    (275,300,250,360,515,540,320,270,350)

qqplot(count_of_items_sold,Unit_price)
```

R code Exa 2.15 Histogram

```
Unit_price = c(40,43,47,74,75,78,115,117,120)
count_of_items_sold =c
   (275,300,250,360,515,540,320,270,350)

hist(Unit_price,breaks = seq(0, 800, by = 10))
```

R code Exa 2.18 Dissimilarity between binary attributes

```
Ja <- c(1,1,0,1,0,0,0)

Jim <- c(1,1,1,0,0,0,0)

ma <- c(0,1,1,1,0,1,0)



Ja_Jim = (1+1)/(1+1+1)
print("Distance between Jack and Jim")

print(Ja_Jim)


Ja_ma =(0+1)/(2+0+1)
print("Distance between Jack and mary")

print(Ja_ma)

Jim_ma =(1+2)/(1+1+2)
print("Distance between Jim and mary")
print(Jim_ma)
```

R code Exa 2.19 Euclidean distance and manhattan distance

```
x1 <- c(1,2)

x2 <- c(3,5)

dif <- x2-x1

Euclidean<- sqrt(sum(dif^2))
print("Euclidean distance")
print(Euclidean)


print("Manhattan Distance")
Manhattan <- sum(dif)

print(Manhattan)
```

R code Exa 2.20 Supremum distance

```
x1 <- c(1,2)

x2 <- c(3,5)

Supremum_Dis<- max(x2)-max(x1)

print("supremum Distance")

print(Supremum_Dis)
```

R code Exa 2.21 Dissimilarity between ordinal attributes

```
test2 = c(3,1,2,3)

obj_Id <- c(1,2,3,4)


dif <- obj_Id - test2

Euclidean<- sqrt(sum(dif^2))
print("Euclidean distance")
print(Euclidean)




test2 = c(1,3)

obj_Id <- c(2,4)


dif <- obj_Id - test2

Euclidean<- sqrt(sum(dif^2))
print("Euclidean distance object 2 and 4")
print(Euclidean)
```

```
test2 = c(3,3)

obj_Id <- c(1,4)


dif <- obj_Id - test2

Euclidean<- sqrt(sum(dif^2))
print("Euclidean distance of object 1 and 4")
print(Euclidean)
```

R code Exa 2.23 cosine similarity between two term frequency vectors

```
x <- c(5,0,3,0,2,0,0,2,0,0)
y <- c(3,0,2,0,1,1,0,1,0,1)



x_square<-sqrt(5^2+0^2+3^2+0^2+2^2+0^2+0^2+2^2+0^2+0^2)



y_square<-sqrt(3^2+0^2+2^2+0^2+1^2+1^2+0^2+1^2+0^2+1^2)




Consine <- ((sum(x*y))/(x_square*y_square))


##### Text Book answer is 0.94
print(Consine)
```

Chapter 3

Data Preprocessing

R code Exa 3.1 Correln analysis of nominal attributes using chi2

```
Obs_fre <- c(250,50,200,1000)

Exp_fre <-c(90,210,360,840)

chi = sum((Obs_fre - Exp_fre)^2/(Exp_fre))


print(chi)
```

R code Exa 3.2 Covariance analysis of numeric attribute

```
AllEle <- c(6,5,4,3,2)

Hightech <-c(20,10,14,5,5)


E_AllEle <- sum(AllEle)/length(AllEle)

```

```
All<- paste("$",E_AllEle)
print(All)
E_Hightech <- sum(Hightech)/length(Hightech)
hi <-paste("$",E_Hightech)
print(hi)
print("Covariance")
cov<- (sum(AllEle*Hightech)/length(AllEle))- (4*E_Hightech)
print(cov)
```

R code Exa 3.3 Histograms

```
AllEle <- c(1,1,5,5,5,5,5,8,8,10,10,10,10,12,14,14,14,15,15,15,15,15,15,18,1
hist(AllEle,main="Histogram for price", xlab="Price", ylab= "Count")
```

R code Exa 3.4 Min Max normalization

```
Min <- 12000
Max <- 98000
Tra <- 73600


Min_max_nor <- (Tra-Min)/(Max-Min)

print("Min-Max Normalization")
print(Min_max_nor)
```

R code Exa 3.5 Z score normalization

```
Mean <- 54000
std <- 16000
Tra <- 73600


Z_score_nor <- (Tra-Mean)/(std)

print("Z-score Normalization")
print(Z_score_nor)
```

R code Exa 3.6 Decimal Scalling

```
decscale<- function (x)
{
    vect <- apply(abs(x), 2, max)
    zvect <- ceiling(log10(vect))
    sc_fact <- 10^zvect
    scale(x, center = TRUE, scale = sc_fact)
}
```

```
print(decscale(iris[,1:4]))
```

Chapter 4

Data warehousing and online analytical processing

R code Exa 4.3 Fact Constellation

```


# Setup the dimension tables


Citytab <- data.frame(key=c("MY", "Ben", "TU", "HU",
    "GU"),
                            name=c("MYSORE", "
                               Bengaluru", "Tumkur", "
                               Hubballi", "Gulabarga")
                               ,
                            country=c("India", "India"
                               , "India", "India", "
                               India"))


weektab <- data.frame(key=1:7,
                            desc=c("Mon", "Tue", "Wen"
                               , "Thu", "Fri", "Sat",
```

```
"Sun"))


prodtab <- data.frame(key=c("Dal", "Sugar", "Rice"), price=c(50, 70, 40))



# Function to generate the Sales table


Totalsales <- function(Record_Size) {


    location <- sample(Citytab$key, Record_Size, replace=T, prob=c(2,2,1,1,1))

    week<- sample(weektab$key, Record_Size, replace=T)

    year <- sample(c(2017,2018), Record_Size, replace=T)

    product <- sample(prodtab$key, Record_Size, replace=T, prob=c(1, 5, 7))

    sales <- data.frame(week=week, year=year, location=location, prod=product)
}


 # create  fact table of sales
Table_fact_sales <- Totalsales(100)

print(Table_fact_sales)
```

R code Exa 4.4 OLAP operations

```


# Setup the dimension tables


Citytab <- data.frame(key=c("MY", "Ben", "TU", "HU", "GU"),
                              name=c("MYSORE", "Bengaluru", "Tumkur", "Hubballi", "Gulabarga"),
                              country=c("India", "India", "India", "India", "India"))


weektab <- data.frame(key=1:7,
                              desc=c("Mon", "Tue", "Wen", "Thu", "Fri", "Sat", "Sun"))


prodtab <- data.frame(key=c("Dal", "Sugar", "Rice"), price=c(50, 70, 40))



# Function to generate the Sales table


Totalsales <- function(Record_Size) {

```

```
    location <- sample(Citytab$key, Record_Size,
      replace=T, prob=c(2,2,1,1,1))

    week<- sample(weektab$key, Record_Size, replace=
      T)

    year <- sample(c(2017,2018), Record_Size,
      replace=T)

    product <- sample(prodtab$key, Record_Size,
      replace=T, prob=c(1, 3, 2))

    sales <- data.frame(week=week, year=year,
      location=location, product=product)
}


 # create  fact table of sales
Table_fact_sales <- Totalsales(20)

print(Table_fact_sales)




Income <- tapply(Table_fact_sales$year, Table_fact_
   sales[,c("product", "week", "year")], FUN=
   function(x){return(sum(x))})


print("Showing the cells of income")

print(Income)


print("Slice")

```

```r
slice<- Income["Dal", "1",]
print(slice)
print("Roll up")
print(apply(Income, c("week", "year"), FUN=function(
    x) {return(sum(x, na.rm=TRUE))}))
print("Drill down")
print(apply(Income, c("week", "year", "product"),
    FUN=function(x) {return(sum(x, na.rm=TRUE))}))
print("piot")
print(apply(Income, c("week", "year"), FUN=function(
    x) {return(sum(x, na.rm=TRUE))}))
print(apply(Income, c("week", "product"), FUN=
    function(x) {return(sum(x, na.rm=TRUE))}))
print("Dice")
print(Income[,c("1","2"),c("2017","2018")])
```

R code Exa 4.6 A data cube is a lattice of cuboids

```r
library(sqldf)
# dimension tables
```

```r

Citytab <- data.frame(key=c("MY", "Ben", "TU", "HU", "GU"),
                              name=c("MYSORE", "Bengaluru", "Tumkur", "Hubballi", "Gulabarga"),
                              country=c("India", "India", "India", "India", "India"))


weektab <- data.frame(key=1:7,
                              desc=c("Mon", "Tue", "Wen", "Thu", "Fri", "Sat", "Sun"))


prodtab <- data.frame(key=c("Dal", "Sugar", "Rice"), price=c(50, 70, 40))



# Function to generate the Total Sales


Totalsales <- function(Record_Size) {


    location <- sample(Citytab$key, Record_Size, replace=T, prob=c(2,2,1,1,1))

    week<- sample(weektab$key, Record_Size, replace=T)

    year <- sample(c(2017,2018), Record_Size, replace=T)

```

```
    product <- sample(prodtab$key, Record_Size, replace=T, prob=c(1, 3, 2))

    sales <- data.frame(week=week, year=year, location=location, product=product)
}


# create  fact table of sales
Table_fact_sales <- Totalsales(20)

###print(Table_fact_sales)

print("Selecting Mysore location")

sel<- sqldf("select * from Table_fact_sales where location = 'MY'")

print(sel)
```

R code Exa 4.8 Join Index

```
library(sqldf)

DataFrame <- data.frame(Seq = rep(10:20, each = 5), tra = rep(1:11,5))

SelQue <- sqldf("select Seq, tra from DataFrame
natural join (select Seq, avg(tra) as avg_tra from DataFrame group by Seq)
where tra> avg_tra")


print(SelQue)
```

R code Exa 4.9 OLAP query processing

```
library(sqldf)

# dimension tables


Citytab <- data.frame(key=c("MY", "Ben", "TU", "HU", "GU"),
                          name=c("MYSORE", "Bengaluru", "Tumkur", "Hubballi", "Gulabarga"),
                          country=c("India", "India", "India", "India", "India"))


weektab <- data.frame(key=1:7,
                          desc=c("Mon", "Tue", "Wen", "Thu", "Fri", "Sat", "Sun"))


prodtab <- data.frame(key=c("Dal", "Sugar", "Rice"), price=c(50, 70, 40))



# Function to generate the Total Sales


Totalsales <- function(Record_Size) {

```

```r

    location <- sample(Citytab$key, Record_Size, replace=T, prob=c(2,2,1,1,1))

    week<- sample(weektab$key, Record_Size, replace=T)

    year <- sample(c(2017,2018), Record_Size, replace=T)

    product <- sample(prodtab$key, Record_Size, replace=T, prob=c(1, 3, 2))

    sales <- data.frame(week=week, year=year, location=location, product=product)
}


 # create  fact table of sales
Table_fact_sales <- Totalsales(20)

###print(Table_fact_sales)

print("Selecting items group by Product")

sel<- sqldf("select * from Table_fact_sales group by product")

print(sel)
```

Chapter 5

Data Cube Technology

R code Exa 5.9 Construct the inverted index

```
A <- c("a1","a1","a1","a2","a2")

B <- c("b1","b2","b2","b1","b1")

C <- c("c1","c1","c1","c1","c1")


D <- c("d1","d2","d1","d1","d1")


E <- c("e1","e1","e2","e2","e3")

#####TID List##########

print("TID List of a1")
print(which("a1" == A))

print("TID List of a2")
print(which("a2" == A))

print("TID List of b1")
```

```
print(which("b1" == B))


print("TID List of b2")
print(which("b2" == B))




print("TID List of c1")
print(which("c1" == C))

print("TID List of d1")
print(which("d1" == D))



print("TID List of e1")
print(which("e1" == E))


print("TID List e2")
print(which("e2" == E))


print("TID List of e3")
print(which("e3" == E))

##### List Size###############

a1 <-length(grep("a1", A))

print("List size of a1")
print(a1)

a2 <-length(grep("a2", A))

print("List size of a2")
```

```
print(a2)

b1 <-length(grep("b1", B))


print("List size of b1")
print(b1)

b2 <-length(grep("b2", B))


print("List size of b2")
print(b2)


c1 <-length(grep("c1", C))


print("List size of c1")
print(c1)


d1 <-length(grep("d1", D))


print("List size of d1")
print(d1)




d2 <-length(grep("d2", D))


print("List size of d2")
print(d2)
```

```
e1 <-length(grep("e1", E))


print("List size of e1")
print(e1)



e2 <-length(grep("e2", E))


print("List size of e2")
print(e2)



e3 <-length(grep("e3", E))


print("List size of e3")
print(e3)
```

R code Exa 5.10 Compute shell fragment

```
A <- c("a1","a1","a1","a2","a2")

B <- c("b1","b2","b2","b1","b1")

C <- c("c1","c1","c1","c1","c1")


D <- c("d1","d2","d1","d1","d1")


```

```
E <- c("e1","e1","e2","e2","e3")

#####TID List##########

print("Cuboid of AB")

print("TID List of a1 and b1")
M1 <- which("a1" == A)
M2<- which("b1" == B)

z1 <- Reduce(intersect, list(M1,M2))

print(length(z1))


print("List size:")
print(length(z1))


print("TID List of a1 and b2")

M3 <- which("a1" == A)
M4<- which("b2" == B)

z2 <- Reduce(intersect, list(M3,M4))
print(z2)

print("List size:")
print(length(z2))


print("TID List of a2 and b1")

M5 <- which("a2" == A)
M6<- which("b1" == B)

z3 <- Reduce(intersect, list(M5,M6))

```

```
print(z3)

print("List size:")
print(length(z3))

print("TID List of a2 and b2")

M7 <- which("a2" == A)
M8<- which("b2" == B)

z4 <- Reduce(intersect, list(M7,M8))

print(z4)

print("List size:")
print(length(z4))



###########################



print("Cuboid of DE")

print("TID List of d1 and e1")
M11 <- which("d1" == D)
M12<- which("e1" == E)

z11 <- Reduce(intersect, list(M11,M12))

print(length(z11))


print("List size:")
print(length(z11))


```

```
print("TID List of d1 and e2")
M13 <- which("d1" == D)
M14<- which("e2" == E)

z12 <- Reduce(intersect, list(M13,M14))

print(length(z12))


print("List size:")
print(length(z12))



print("TID List of d1 and e3")
M15 <- which("d1" == D)
M16<- which("e3" == E)

z13 <- Reduce(intersect, list(M15,M16))

print(length(z13))


print("List size:")
print(length(z13))


print("TID List of d2 and e1")
M17 <- which("d2" == D)
M18<- which("e1" == E)

z14 <- Reduce(intersect, list(M17,M18))

print(length(z14))


print("List size:")
print(length(z14))
```

R code Exa 5.11 Computing cubes with average measure

```
print(cbind(TID=c(1,2,3,4,5),Item_count=c(5,3,8,5,2)
    ,SUM=c(70,10,20,40,30)))
```

Chapter 6

Mining frequent patterns associations and correlations basic concepts and methods

R code Exa 6.9 Correlation analysis using chi2

```
library(arules)
Video <- c(4000,2000,3500,500)
Video_game <-c(4500,1500,3000,1000)

cor <- sum((Video-Video_game)^2/Video_game)


print(cor)
```

Chapter 7

Advanced Pattern mining

R code Exa 7.1 Mining multilevel association rules

```
## Taken from arules package PDF
library(arules)

data("Groceries")

## Groceries contains a hierarchy stored in itemInfo

Groceries_level2 <- aggregate(Groceries, by = "
   level2")

inspect(Groceries_level2)
```

R code Exa 7.2 redundancy among multilevel association rules

```
library(arules)

#### it demonstartes redundant rules and constraints
    (Example 7.8)
```

```
data("Income")



Ass_rules <- apriori(Income, parameter = list(
    support = 0.5,conf=0.9))


inspect(rules[is.redundant(Ass_rules)])
```

R code Exa 7.3 Rare patterns and negative patterns

```
library(arules)


### No rare Items in the dataset so I am showing
    other measure like significance of rules


data("Income")



Ass_rules <- apriori(Income, parameter = list(
    support = 0.5,conf=0.9))


interestMeasure(Ass_rules,measure = "hyperConfidence
    ",transactions = Income)
```

R code Exa 7.6 Negatively related patterns

```r
library(arules)


### Complement Items from the dataset



data("Adult")
rules <- apriori(Adult)

InMe<- interestMeasure(rules, measure = "kulczynski"
    ,transactions = Adult)


print(InMe)
```

R code Exa 7.12 closed and maximal itemsets

```r
library(arules)

data("Adult")

## find only frequent itemsets which do not contain
    small or large income

items <- apriori(Adult, parameter = list(support=
    0.001, conf=0.001, target="frequent"))

close <-is.closed(items)


print(close)
```

R code Exa 7.13 pattern distance

```
library(arules)

data("Adult")

sample <- sample(Adult, 10)

## used Jaccard distance

jaccard_dist <- dissimilarity(sample)

hira_clu <- hclust(jaccard_dist , method = "ward.D2"
   )

plot(hira_clu, labels = FALSE, main = "Dendrogram")
```

R code Exa 7.15 Semantic annotations of a frequent patterns

```
library(arules)

data("Adult")

sample <- sample(Adult, 100)


sample1 <- apriori(Adult[1:50], parameter = list(
   support = 0.6))
sample2 <- apriori(Adult[51:100], parameter = list(
   support = 0.6))


combine_samples <- c(sample1, sample2)


```

```
print(duplicated(combine_samples))
```

Chapter 8

Classification basic concepts

R code Exa 8.9 Sensitivity and Specificity

```
TP <- 90

FN <-210

FP <- 140

TN <- 9560
```

```
Sensitivity= TP/(TP+FN)

print("Sensitivity")

print(Sensitivity)

Specificity = TN/(FP+TN)

print("Specificity")

print(Specificity)
```

R code Exa 8.10 Precision and recall

```
TP <- 90

FN <-210

FP <- 140

TN <- 9560


Precision = TP/(TP+FP)

print("Precision")

print(Precision)

Recall = TP/(TP+FN)

print("Recall")

print(Recall)
```

R code Exa 8.11 ROC Curve

```
library(ROCR)

dataset<-data.frame(Pre=c
    (0.90,0.80,0.70,0.60,0.55,0.54,0.53,0.51,0.50,0.40)
    ,cls=c(1,1,0,1,1,0,0,0,1,0))


predictions <- prediction(dataset$Pre, dataset$cls)

model_perf <- performance(predictions,"tpr","fpr")

plot(model_perf)
```

Chapter 9

Classification Advanced methods

R code Exa 9.1 Backpropagation algorithm

```
### install.packages("neuralnet")


library(neuralnet)
library(MASS)
data <- Boston


## used learning rate 0.9 and one hidden layer




print(net.infert <- neuralnet(medv~nox+rm+age,
   learningrate = 0.9,data,hidden=1,act.fct="tanh"))



prediction(net.infert)
```

R code Exa 9.3 Error correcting codes

```
library(e1071)


C1 <- c(1,1,1,1,1,1,1)
C2 <- c(0,0,0,0,1,1,1)
C3 <- c(0,0,1,1,0,0,1)
C4 <- c(0,1,0,1,0,1,0)

out <-c(0,0,0,1,0,1,0)

Out1<-hamming.distance(C1, out)

print(Out1)


Out2<-hamming.distance(C2, out)

print(Out2)

Out3<-hamming.distance(C3, out)

print(Out3)


Out4<-hamming.distance(C4, out)

print(Out4)
```

Chapter 10

Cluster analysis basic concepts and mehtods

R code Exa 10.1 Clustering by K means partitioning

```

data <- iris
data$Species <- NULL

clusters <- kmeans(data, 3)

plot(data[c("Sepal.Length", "Sepal.Width","Petal.
   Length", "Petal.Width")], col=clusters$cluster)
```

R code Exa 10.2 Drawback of k means

```

data <- c(1,2,3,8,9,10,25)

clu1 <- c(1,2,3)

```

```
clu2 <- c(8,9,10,25)


Mean_clu1 <- mean(clu1)

Mean_clu2 <- mean(clu2)


su_mean1 <- sum((clu1-Mean_clu1)^2)

su_mean2 <- sum((clu2-Mean_clu2)^2)


First_Total<-su_mean1+su_mean2

print("Variation within first partition")

print(First_Total)




print("
   ##################################################################
   ")



data <- c(1,2,3,8,9,10,25)

clu3 <- c(1,2,3,8)

clu4 <- c(9,10,25)


Mean_clu3 <- mean(clu3)

Mean_clu4 <- mean(clu4)
```

```
su_mean3 <- sum((clu3-Mean_clu3)^2)
su_mean4 <- sum((clu4-Mean_clu4)^2)
sec_Total<-su_mean3+su_mean4
print("Variation within second partition")
print(sec_Total)
```

R code Exa 10.3 Agglomerative versus divisive hierachical clustering

```
 library(cluster)
data(iris)
print("Agglomerative Clustering")
agn_hiclu <- agnes(iris, metric = "manhattan", stand
    = TRUE)
print(agn_hiclu)
plot(agn_hiclu)
print("
   #######################################################
   ")
data(iris)
```

```
print("Devisive Clustering")
divisive_clu <- diana(iris, metric = "manhattan",
    stand = TRUE)
print(divisive_clu)
plot(divisive_clu)
```

R code Exa 10.4 Single versus complete linkages

```
index <- sample(1:dim(iris)[1], 60)
newiris <- iris[index,]
newiris$Species <- NULL

###Apply Hierarchical Clustering

hier_clu <- hclust(dist(newiris), method="ave")
plot(hier_clu , hang = -1, labels=newiris$Species[
    index])
```

R code Exa 10.7 Density reachability and density connectivity

```
library(dbscan)

data(iris)

iris <- as.matrix(iris[,1:4])

result_dbscan <- dbscan(iris, eps = .3, minPts = 3)


print(result_dbscan)
```

```
pairs(iris, col = result_dbscan$cluster + 1L)
```

R code Exa 10.8 core distance and reachability distance

```

library(dbscan)

data(iris)



result <- optics(iris[,1:4], eps = 10, minPts = 5)


###Componets of reachability
Com_reach <- as.reachability(result)


###plot(Com_reach, order_labels = TRUE)


dend <- as.dendrogram(Com_reach)


plot(dend)
```

Chapter 11

Advanced cluster analysis

R code Exa 11.5 Probabilistic clusters

```
library(FPDclustering)


Pro_clu <- PDclust(iris[,1:4], k = 2)


print(Pro_clu)
```

R code Exa 11.7 Fuzzy clustering using the EM algorithm

```
###Install package install.packages("EMCluster")


library(EMCluster)
library(MASS)
library(Matrix)

```

```
Dataset<- data.frame(Fi=c(4,3,9,14,18,21),se=c
    (10,3,6,8,11,7))


d <- as.matrix(Dataset)

emobj <- simple.init(d, nclass = 2)
emobj <- shortemcluster(d, emobj)

emclu <- emcluster(d, emobj, assign.class = TRUE)
print(emclu)
```

R code Exa 11.14 clustering in a derived space

```
##install.packages("speccalt")


library(speccalt)
iris <- local.rbfdot(iris[,1:4])
cluster1 <- speccalt(iris) # with automatic
    estimation
cluster2 <- speccalt(iris, 4)


View(cluster1)

View(cluster2)
```

R code Exa 11.16 Bipartite graph

```
###install.packages("igraph")
```

```
library(igraph)

graph <- make_full_bipartite_graph(2, 2, dir=TRUE,
   mode="all")


print(graph, v=TRUE)


plot(graph)
```

R code Exa 11.19 Measurements based on geodesic distance

```


library(geosphere)
#geodesic distance
geo_dest<- geodesic(cbind(0,0), 2, 3)

print(geo_dest)
```

R code Exa 11.21 cuts and clusters

```

hc <- hclust(dist(iris[,1:4]))


cutree(hc, k = 1:3) #k = 1 is trivial
cutree(hc, h = 100)

## Compare the 2 and 10 grouping:
gra210 <- cutree(hc, k = c(2,10))
```

```r


plot(gra210)
```

R code Exa 11.23 Hard and soft constraints

```r

library(SoftClustering)

data(iris)

Hardclu <- HardKMeans(iris[,1:4],2,2,10)


print(Hardclu)
```

Chapter 12

Outlier detection

R code Exa 12.1 Outliers

```
library(outliers)



outliers<- outlier(iris[,1:4])

print(outliers)
```

R code Exa 12.7 Detecting outliers using clustering

```
iris <- iris[,1:4]


kmeansClu <- kmeans(iris, centers=3)



centersofclu <- kmeansClu$centers[kmeansClu$cluster,
    ]
```

```
dist <- sqrt(rowSums((iris - centersofclu)^2))

outliers <- order(dist, decreasing=T)[1:10]

print(outliers)
```

R code Exa 12.8 Univariate outliers detection using maxumum likelihood

```
dataset <- c
    (24,28.9,28.9,29,29.1,29.1,29.2,29.2,29.3,29.4)

mean(dataset)
print(sd(dataset))
```

R code Exa 12.9 Multivariate outlier detection using mahalanobis distance

```

library(mvoutlier)

data(iris)


aq.plot(iris[,1:4], alpha=0.1)
```

R code Exa 12.10 Multivariate outlier detection using the chi2 statistic

```

library(outliers)

```

```
dataset <- c
   (24,28.9,28.9,29,29.1,29.1,29.2,29.2,29.3,29.4)
chisq.out.test(dataset )
print(chisq.out.test(dataset, opposite=TRUE))
```

R code Exa 12.12 Multivariate outlier detection using multiple clusters

```
library(kmodR)

d<- as.matrix(iris[,1:4])

print(kmod(d,k=3,l=10, i_max = 100))
```

R code Exa 12.13 Outlier detection using a histogram

```


d<- iris[,1:4]

hist(as.matrix(d))
```

R code Exa 12.14 local proximity based outliers

```
library(DMwR)



d<- iris[,1:4]

local_pro<- lofactor(d[,-5],10)
```

```
print(local_pro)
```

R code Exa 12.15 Detecting outliers as objects that do not belong to any clusters

```
library(DMwR)
library(lattice)
library(grid)



d11<- iris[,1:4]

d <- as.matrix(d11)

density_scan<- dbscan(d, eps=1, minPts = 5)


print(density_scan)
```

R code Exa 12.16 clustering based outliers detection using distance to the closest cluster

```
library(DMwR)
library(lattice)
library(grid)



dataset<- iris[,1:4]

```

```
d <- as.matrix(dataset)

dist<- kNNdist(d, k=4, search="kd")

print(dist)


kNNdistplot(d, k=4)
```

R code Exa 12.18 detecting outliers in small clusters

```
library(DMwR)
 # Ignore class column "Species", which is a
    categorical column

iris <- iris[,1:4]

outlierslof <- lofactor(iris, k=2)

outliers <- order(outlierslof, decreasing=T)[1:10]

print(outliers)
```

R code Exa 12.20 Outlier detection by semi supervised learning

```
library(ldbod)

dataset <- as.matrix(iris[,1:4])

local_den<- ldbod(dataset, k=3, nsub=50)


print(local_den)
```

R code Exa 12.24 Outliers in subspace

```r
library(HighDimOut)
library(ggplot2)

result_SOD <- Func.SOD(data = iris[,1:4], k.nn = 10,
    k.sel = 5, alpha = 0.8)

plot(result_SOD)
```

R code Exa 12.25 angle based outliers

```r


###install.packages("abodOutlier")


library(abodOutlier)

data(iris)

Abodf<- abod(iris[,1:4], method = "randomized", n_
   sample_size = 5)


View(Abodf)
```