Activity Sheet

Learning outcomes:

After solving these exercises, you should be able to understand the following concepts.

a. Reading transaction data and exploring the data and items

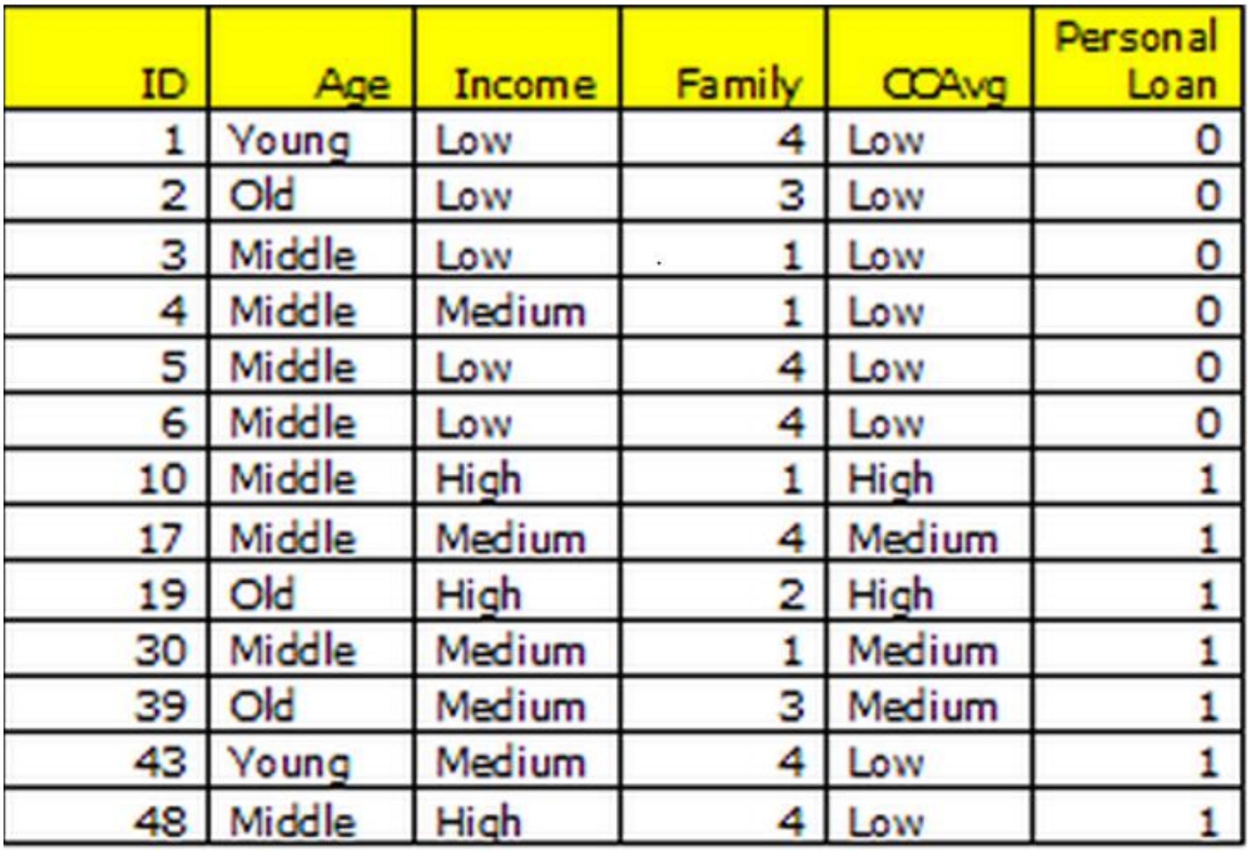
b. Implementing association rule mining in R

c. Understanding the computation of support, confidence and lift

d. Interpreting rules and results

e. Obtaining the patterns/rules from a supervised dataset and computing the metrics support, confidence, lift for the rules

1. Based upon the dataset calculate the support, confidence and lift for the following rule "if loan = accept then CCAvg is medium"



ID	Age	Income	Family	CCAvg	Personal Loan
1	Young	Low	4	Low	0
2	Old	Low	3	Low	0
3	Middle	Low	1	Low	0
4	Middle	Medium	1	Low	0
5	Middle	Low	4	Low	0
6	Middle	Low	4	Low	0
10	Middle	High	1	High	1
17	Middle	Medium	4	Medium	1
19	Old	High	2	High	1
30	Middle	Medium	1	Medium	1
39	Old	Medium	3	Medium	1
43	Young	Medium	4	Low	1
48	Middle	High	4	Low	1

2. Association Rules for transaction data:

Steps to Follow:

a. Install and load ‘arules’ package

```
install.packages("arules")

 library(arules)
```

b. Read transaction data using 'Transactions.csv' arules package function

```
trans = read.transactions(file="Transactions.csv",
          rm.duplicates= FALSE,
           format="single", sep=",",
          cols =c(1,2))
```

c. Check the data read format

```
inspect(trans)
```

d. Explore and understand the data and items of transaction data

```
trans
```

e. Find itemFrequency and plot the same

```
itemFrequency(trans)

itemFrequencyPlot(trans)
```

f. Implementing association mining using ‘Apriori’ algorithm to extract rules

```
rules <- apriori(trans,parameter = list(sup = 0.2, conf = 0.6,target="rules"))
```

g. Inspect the rules

```
inspect(rules)
```

h. Order of rules in decreasing order of confidence and support

```
rules = as(rules[sort(rules, by = c("confidence", "support"), order = TRUE)],
 "data.frame")
```

i. Write the rules to a file

```
require(stringr)

 m = str_split(rules$rules,"=>")

 rhs = data.frame(RHS = unlist(lapply(m,function(x){str_trim(x[2])})))
 lhs = data.frame(LHS = unlist(lapply(m,function(x){str_trim(x[1])})))
 rules_csv = data.frame(lhs, rhs, rules[,c("support", "confidence", "lift")])
 rules_csv = unique(rules_csv)

write.csv(rules_csv, "Rules.csv")
```

3. Association rules for Titanic Data:

Steps to Follow:

a. Load titanic data

```
load("titanic.raw.rdata")
```

b. Understand the data using summary and str

```
 class(titanic.raw)
 summary(titanic.raw)
  str(titanic.raw)
```

c. Select 5 sample records

```
head(titanic.raw,5)

 idx <- sample(1:nrow(titanic.raw), 5)
titanic.raw[idx, ]
```

d. Convert the data into transaction format

```
  inspect(titanic)

  itemFrequency(titanic)
  itemFrequencyPlot(titanic)
```

e. Find association rules with default settings

```
rules.all <- apriori(titanic)
rules.all
inspect(rules.all)
```

f. Generating rules with rhs containing “Survived”

```
 rules <- apriori(titanic.raw,
                control = list(verbose=F),
                parameter = list(minlen=2, supp=0.005, conf=0.8),
                appearance = list(rhs=c("Survived=No",
                                          "Survived=Yes"),
                                   default="lhs"))
```

g. Inspect the rules

```
inspect(rules)
```

h. Keep data only upto 3 decimal places

```
  quality(rules) <- round(quality(rules), digits=3)
```

i. Order rules by lift

```
 rules.sorted <- sort(rules, by="lift")
inspect(rules.sorted)
```

j. Removing redundant rules

```
 ## find redundant rules
subset.matrix <- is.subset(rules.sorted, rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag = T)] <- NA
redundant <- colSums(subset.matrix, na.rm = T) >= 1

## which rules are redundant
which(redundant)

## remove redundant rules
rules.pruned <- rules.sorted[!redundant]

# Remaining Rules
inspect(rules.pruned)
```