Homework- 3 Association Rules  
Shivani S Mahaddalkar

**Analysis Steps:**

1. To apply the association rules we have to import the necessary packages and the dataset into R.

> library(dplyr)

> library(arules)

> library(arulesViz)

>

> bankdata <- read.csv("C:/Users/shiva/Downloads/bankdata\_csv\_all.csv")

> str(bankdata)

'data.frame': 600 obs. of 12 variables:

$ id : chr "ID12101" "ID12102" "ID12103" "ID12104" ...

$ age : int 48 40 51 23 57 57 22 58 37 54 ...

$ sex : chr "FEMALE" "MALE" "FEMALE" "FEMALE" ...

$ region : chr "INNER\_CITY" "TOWN" "INNER\_CITY" "TOWN" ...

$ income : num 17546 30085 16575 20375 50576 ...

$ married : chr "NO" "YES" "YES" "YES" ...

$ children : int 1 3 0 3 0 2 0 0 2 2 ...

$ car : chr "NO" "YES" "YES" "NO" ...

$ save\_act : chr "NO" "NO" "YES" "NO" ...

$ current\_act: chr "NO" "YES" "YES" "YES" ...

$ mortgage : chr "NO" "YES" "NO" "NO" ...

$ pep : chr "YES" "NO" "NO" "NO" ...

1. The ID column which was not needed for the analysis was dropped.

> bd <- bankdata[-1]

> head(bd)

age sex region income married children car save\_act current\_act mortgage pep

1 48 FEMALE INNER\_CITY 17546.0 NO 1 NO NO NO NO YES

2 40 MALE TOWN 30085.1 YES 3 YES NO YES YES NO

3 51 FEMALE INNER\_CITY 16575.4 YES 0 YES YES YES NO NO

4 23 FEMALE TOWN 20375.4 YES 3 NO NO YES NO NO

5 57 FEMALE RURAL 50576.3 YES 0 NO YES NO NO NO

6 57 FEMALE TOWN 37869.6 YES 2 NO YES YES NO YES

1. The columns that have character strings in the input are modified to turn them into factors for analysis.

> bd$sex <- as.factor(bd$sex)

> bd$region <- as.factor(bd$region)

> bd$married <- as.factor(bd$married)

> bd$children <- as.factor(bd$children)

> bd$car <- as.factor(bd$car)

> bd$save\_act <- as.factor(bd$save\_act)

> bd$current\_act <- as.factor(bd$current\_act)

> bd$mortgage <- as.factor(bd$mortgage)

> bd$pep <- as.factor(bd$pep)

1. Discretizing the age variable and putting into categories :child, teens, young-adult, middle-aged-adult, old-aged-adult, senior)

> bd$age <- cut(bd$age, breaks = c(0,12,18,25,44,60,Inf),labels=c("child","teens","young-adult", "middle-aged-adult" , "old-aged-adult", "senior"))

> bd$age

[1] old-aged-adult middle-aged-adult old-aged-adult young-adult old-aged-adult old-aged-adult young-adult

[8] old-aged-adult middle-aged-adult old-aged-adult senior old-aged-adult middle-aged-adult senior

[15] middle-aged-adult middle-aged-adult middle-aged-adult old-aged-adult senior middle-aged-adult senior

[22] old-aged-adult old-aged-adult middle-aged-adult young-adult old-aged-adult old-aged-adult middle-aged-adult

[29] middle-aged-adult senior senior young-adult old-aged-adult middle-aged-adult middle-aged-adult

1. Discretizing the income column into 4 categories: "lower class", "lower middle class", "upper middle class", "upper class"

> bd$income <- cut(bd$income, 4, labels = c("lower class", "lower middle class", "upper middle class", "upper class"))

> bd$income

[1] lower class lower middle class lower class lower middle class upper class upper middle class lower class

[8] lower middle class lower middle class lower middle class upper class lower middle class lower class upper class

[15] lower class lower middle class lower class upper middle class lower middle class lower middle class upper class

[22] lower class upper middle class lower class lower class upper middle class lower middle class lower middle class

[29] lower middle class lower middle class lower middle class lower class lower middle class lower middle class upper middle class

[36] lower middle class lower class lower class lower class lower middle class lower middle class lower middle class

[43] upper class lower middle class lower class upper class lower class lower class upper middle class

1. Coercing the bank dataframe into a sparse transaction matrix for rule association.  
   > bd <- as(bd,"transactions")
2. Finding the rules with parameters support = 0.04, confidence=0.9, maxlen=3  
   > rules <- apriori(bd, parameter = list(supp = 0.04, conf = 0.9, maxlen=))
3. Inspecting the 28 rules generated.  
   > inspect(rules)
4. Inspecting the rules with pep= ‘YES’ as the rhs

rules1<-apriori(data=bd, parameter=list(supp=0.04,conf = 0.9,minlen=3),

appearance = list(default="lhs",rhs=('pep=YES')),

control = list(verbose=F))

**Insights generated from the analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LHS | RHS | Support | Confidence | Lift |
| young-adult,  save\_act=YES | income=lower class | 0.08666667 | 0.9629630 | 3.009259 |
| income=lower middle class,  children=1 | pep=YES | 0.08333333 | 0.9615385 | 2.105559 |
| age=young-adult,  mortgage=YES | income=lower class | 0.04500000 | 0.9310345 | 2.909483 |
| income=upper class,  current\_act=YES | save\_act=YES | 0.07500000 | 1.0000000 | 1.449275 |
| age=young-adult,car=YES | income=lower class | 0.05666667 | 0.9444444 | 2.9513 |

The values of the above rules suggest that young-adults have a larger probability of lying in the income group of lower class. The same association can be seen strongly in 3 out of the 5 rules. It means that capturing such an client base early could prove beneficial for later. If the PEP plan is adjusted to accommodate the lower income class, the product could get a boost in its popularity. Further research could be done to gauge the responses from young adults who do not have a large disposable income.

In rule number 2, it is evident that people with a lower middle class income and have a child will agree to a PEP plan. It could be determined that people with slightly more disposable income than the lower class, and a responsibility towards the future tend to gravitate more towards the PEP plan. In this case, the lift value of 2.105 suggests that there is a strong association between the LHS and the RHS. It means that the people ended up buying PEP because it has a strong association with lower middle income and having a child. The value of 0.961 suggests that when the income group and people having one child was chosen, majority(96.1%) of people had opted for the PEP plan. The support shows that the probability of all the items occurring together is 0.0833.