MULTIMEDIA UNIVERSITY
TDS 3301 DATA MINING

ASSIGNMENT 2
ASSOCIATION RULE MINING

GROUP DETAILS

NAME	ID	EMAIL
Darrel Shakri Bin Ahmad Shakri	1141327906	menubearer@gmail.com
Nur Nadhirah Bt. Nazarudin	1142700151	nazanadhirah@gmail.com
Andy Yong Jun Jie	1142701565	Andyyong9611@gmail.com
Kirbashini Naidu a/p Ragavelu	1141127226	Kirba4796@gmail.com

Domain

Retail

Potential Benefits from association rule mining

- Manage the position of products in the bakery store. in such a way, to the point that customers can sensibly discover things he/she may purchase which expands the consumer loyalty and subsequently the benefit.

- To diminish the search issue to a more sensible size.

- Reports with respect to expectation of item deals patterns and client customer behaviour. To let retailers to make hands-on, learning driven choices.

Improvement

Build up an application with a plan to observe and measure execution utilizing key execution pointers (KPIs) in light of recorded data, current conditions and future objectives, dissect sales patterns and client purchasing examples to help benefit and make exact figures about future deals.

About the bakery dataset

The extended bakery datasets stores receipts of various amounts and the goods present in each receipts. The goods are from the items table of the bakery datasets, and consists of 50 different items.

There are other data in the dataset as well, such as store locations and employee information.

Pre-processing for market basket analysis

To perform market basket analysis, a binary vector between the receipts and bakery goods bought in the receipts must be made.

Because the dataset is in the form of SQL, construction of the database is necessary, the construction is detailed in *construction.R* on the GitHub page.

When the tables have been constructed, we first extract the names of the goods with the following code:

```
library(sqldf)


db <- dbConnect(SQLite(), dbname="bakery1000")
options(warn=-1)


# read goods table to a dataframe
```

```
goods <- dbGetQuery(db,'select * from goods')
```

With the, the dataframe goods has been extracted. After this process, we extract the names to a vector:

```
# get the list of goods in database
goodsList <- paste(goods$Flavor, goods$Food)
```

This results in the vector of item names in a proper format.

Next, we create an empty vector with row size 1000 (for the receipts) and column size 50 (for the items):

```
# create empty matrix as binary vector
binaryVector <- matrix(0, ncol = nrow(goods), nrow = max(items$Receipt))
```

To fill out the vector, we extract the items table, and use it as an index to fill out the correct locations:

```
# get vector of goods id in item
# must plus one because database ID is zero-indexed
# while R matrices are one-indexed
goodsVector <- as.vector(items$Item + 1)


# combine id and goods vector for indexing
idx <- cbind(idVector,goodsVector)

# setup vector
binaryVector[idx] <- 1

binaryDF <- data.frame(binaryVector)
```

Finally, we insert the goods name vector as column names, and the binary vector is ready for association rule mining:

```
# setup binary vector columns
colnames(binaryVector) <- goodsList
```

Full details on constructing the binary vector can be found on the file *binaryVector.R***,** though you need to perform *construction.R* first in order to build the necessary database tables.

Performing Association Rules Mining

Based on the binary vector, we perform association rule mining based on these parameters:

Parameter settings: minlen=2, support = 0.005, confidence = 0.8
Choice of algorithm : Apriori
Time required: 0.03 seconds

```
> rulesB <- apriori(binaryVector,parameter=list(minlen=2,supp=0.005,conf=0.8))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.8    0.1    1 none FALSE            TRUE       5   0.005      2     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 5

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[50 item(s), 1000 transaction(s)] done [0.00s].
sorting and recoding items ... [50 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [88 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> quality(rulesB)<-round(quality(rulesB),digits=3)
>
```

Execution of apriori with the given parameters

Results

The quality measures are tabulated below:

	Support	**Confidence**	**Lift**
Minimum	0.005	0.80	8.4
Maximum	0.04	1.00	19.608
Mean	0.024	0.937	12.49
Median	0.024	0.95	12.516

The associations gathered have a high level of confidence, allowing us to recommend any of the frequent itemsets discovered. However, due to the relatively low support, usefulness may be limited.

We would recommend the following associations to be used for product placement:

```
    lhs                                     rhs                  support confidence lift
[1] {Apple Tart,Apple Danish}            => {Apple Croissant}    0.040   0.976      10.721
[2] {Apple Tart,Apple Croissant}         => {Apple Danish}       0.040   0.909      10.823
[3] {Apple Croissant,Apple Danish}       => {Apple Tart}         0.040   0.952      12.055
[4] {Chocolate Cake,Casino Cake}         => {Chocolate Coffee}   0.038   0.950      11.176
[5] {Casino Cake,Chocolate Coffee}       => {Chocolate Cake}     0.038   0.974      11.600
[6] {Chocolate Cake,Chocolate Coffee}    => {Casino Cake}        0.038   0.809      11.229
```

Recommended associations

Recommendations

According to the selected rules, the customers tend to buy items of the same or similar flavour. Therefore, the bakery should separate the items by distinct bakery good type and keep items of the same flavour separate so that the customers would have to walk through the entire store to get their normal items, leading to increased chance of looking and purchasing other items.