Université d'Ottawa
Faculté de génie
École de science informatique
et de génie électrique

uOttawa
L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering
School of Electrical Engineering
and Computer Science

Fundamental of Data Science

Assignment Four

Reyad Melies

Part A: Association Rules:

a) Find all frequent item sets in database X.

	A	B	C	D	E	F	G	SUM
A	5	3	3	4	1	2	1	19
B	3	4	2	2	0	1	2	14
C	3	2	5	4	1	2	3	20
D	4	2	4	6	1	4	3	24
E	1	0	1	1	1	0	1	5
F	2	1	2	4	0	4	2	15
G	2	2	3	3	1	2	5	18
SUM	20	14	20	24	5	15	17	115

A	5
B	4
C	5
D	6
E	1
F	4
G	5

b)Find strong association rules for database:

		support
A	5	0.625
B	4	0.5
C	5	0.625
D	6	0.75
E	1	0.125
F	4	0.5
G	5	0.625

A	5
B	4
C	5
D	6
F	4
G	5

0.25*8=2

Records>=2 :Remove E

A,B	3
A,C	3
A,D	4
A,F	2
A,G	2
B,C	2
B,D	2
B,F	1
B,G	2
C,D	4
C,F	2
C,G	3
D,F	4
D,G	3
F,G	2

→

A,B	3
A,C	3
A,D	4
A,F	2
A,G	2
B,C	2
B,D	2
B,G	2
C,D	4
C,F	2
C,G	3
D,F	4
D,G	3
F,G	2

Records>=2

Remove {B,F}

A,B,C	1
A,B,D	2
A,C,D	3
A,C,F	1
A,B,G	1
A,C,G	1
A,D,F	2
A,D,G	1
A,F,G	0
B,C,D	1
B,C,G	1
B,D,G	0
C,D,F	2
C,D,G	2
C,F,G	1
D,F,G	2

→

A,B,D	2
A,C,D	3
A,D,F	2
C,D,F	2
C,D,G	2
D,F,G	2

Remove { A,B,C },{ A,C,F},{ A,B,G },{ A,C,G },{ A,D,G },{ A,F,G },{ B,C,D},
{ B,C,G },{ B,D,G },{ C,F,G }

Rule Generation:

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

$$\text{lift}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) \times \text{supp}(Y)}$$

A,B->D	"2/3"	66.6667
A,D->B	"2/4"	50
B,D->A	"2/2"	100
A->A,B	"2/5"	40
B->A,D	"2/4"	50
D->A,B	"2/6"	33.3333
A,C->D	"3/3"	100
A,D->C	"3/4"	75
C,D->A	"3/4"	75
A->C,D	"3/4"	75
C->A,D	"3/5"	60
D->A,C	"3/6"	50
C,D->F	"2/4"	50
C,F->D	"2/2"	100
F,D->C	"2/4"	50
C->D,F	"2/5"	40
D->C,F	"2/6"	33.333
F->C,D	"2/4"	50
C,D->G	"2/4"	50
C,G->D	"2/2"	100
G,D->C	"2/3"	66.66667
C->D,G	"2/5"	40
D->C,G	"2/6"	33.3333
G->C,D	"2/5"	40
D,F->G	"2/4"	50
D,G->F	"2/3"	66.6667
F,G->D	"2/2"	100
D->F,G	"2/6"	33.333
F->D,G	"2/4"	50
G->D,F	"2/5"	40
A,D->F	"2/4"	50
A,F->D	"2/2"	100
DF->A	"2/4"	50
A->F,D	"2/5"	40
F->A,D	"2/4"	50
D->A,F	"2/6"	33.3333

Choose confidence >=60

		Confidence	Lift
A,B->D	"2/3"	66.6667	0.889
B,D->A	"2/2"	100	1.6
A,C->D	"3/3"	100	1.3
A,D->C	"3/4"	75	1.2
C,D->A	"3/4"	75	1.2
A->C,D	"3/5"	60	1.2
C->A,D	"3/5"	60	1.2
C,F->D	"2/2"	100	1.3
C,G->D	"2/3"	66.67	0.8889
G,D->C	"2/3"	66.666667	1.0667
D,G->F	"2/3"	66.6667	1.3
F,G->D	"2/2"	100	1.3
A,F->D	"2/2"	100	1.3

We have 13 rules driven.

b) Analyze misleading associations for the rule set obtained in (b).

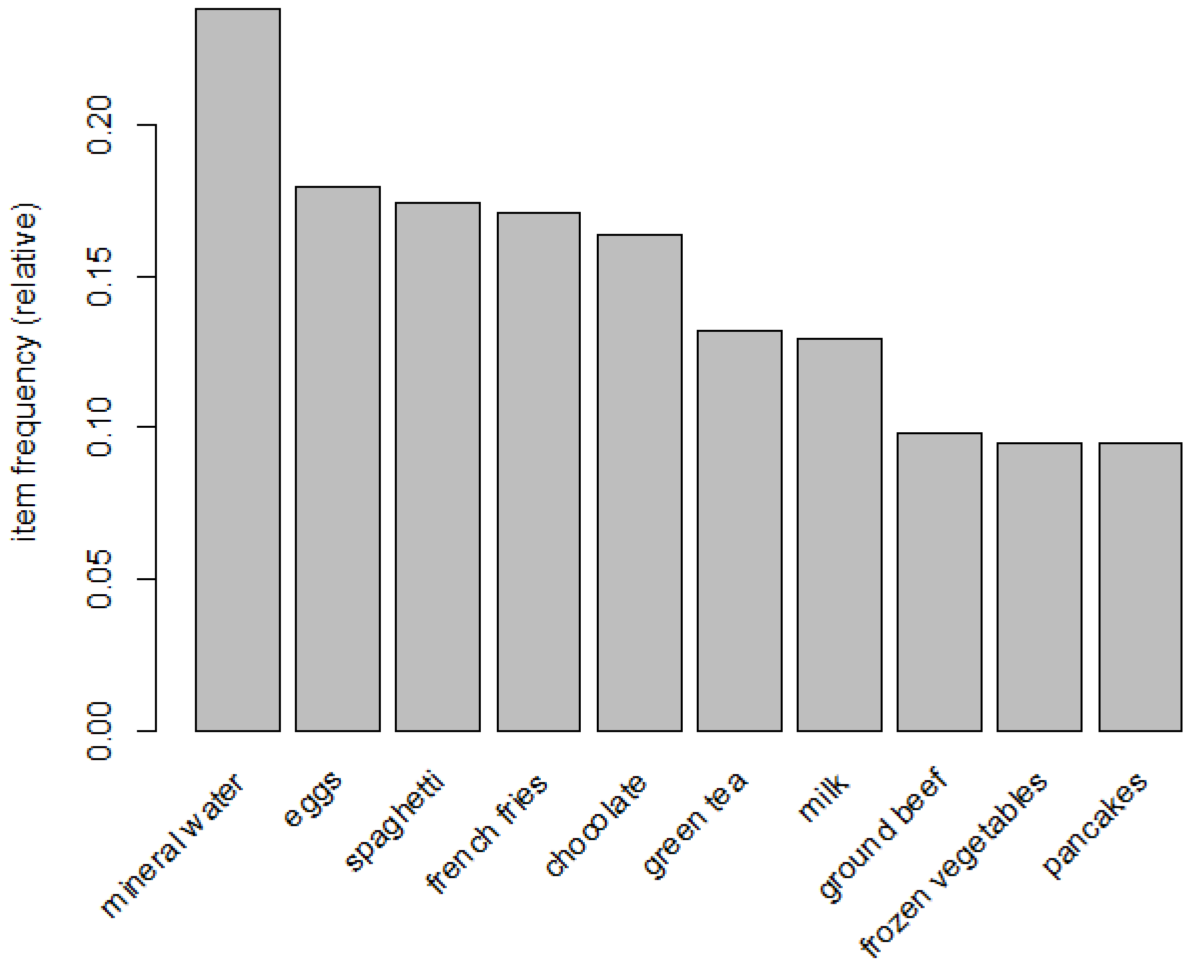
Therefore, we calculated lift. Events with high probability is likely to happen events related to event happening or not and can be even negatively associated. As Support & confidence are insufficient at filtering out uninteresting rules.

Calculating lift values which are less than 1 are misleading values.

A,B->D	"2/3"	66.6667	0.889
C,G->D	"2/3"	66.67	0.8889

The rest are valid rules.

II. a) Generate a plot of the top 10 transactions.



b) • Generate association rules using minimum support of 0.002, minimum confidence of 0.20, and maximum length of 3.

```
model <- apriori(dataset, parameter = list(support =
                                            0.002, confidence = 0.20, maxlen = 3))
model
# summary of grocery association rules
summary(model)
```

```
> summary(model)
set of 2023 rules

rule length distribution (lhs + rhs):sizes
   1    2    3
   1  357 1665

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   3.000   2.823   3.000   3.000

summary of quality measures:
    support           confidence        coverage             lift             count
 Min.   :0.002133   Min.   :0.2000   Min.   :0.002666   Min.   : 0.8595   Min.   :  16.0
 1st Qu.:0.002533   1st Qu.:0.2405   1st Qu.:0.008266   1st Qu.: 1.5377   1st Qu.:  19.0
 Median :0.003466   Median :0.2941   Median :0.011465   Median : 1.8674   Median :  26.0
 Mean   :0.005292   Mean   :0.3177   Mean   :0.018647   Mean   : 2.0415   Mean   :  39.7
 3rd Qu.:0.005599   3rd Qu.:0.3774   3rd Qu.:0.019064   3rd Qu.: 2.3381   3rd Qu.:  42.0
 Max.   :0.238368   Max.   :0.9500   Max.   :1.000000   Max.   :28.0881   Max.   :1788.0

mining info:
    data ntransactions support confidence
 dataset          7501   0.002        0.2
```

Display the rules, sorted by descending lift value.

```
# sorting model rules by lift to determine actionable rules
inspect(sort(model, by = "lift"))
```

```
     lhs                                   rhs                      support     confidence coverage    lift      count
[1]  {escalope,mushroom cream sauce}     => {pasta}                 0.002532996 0.4418605 0.005732569 28.088096 19
[2]  {escalope,pasta}                    => {mushroom cream sauce}  0.002532996 0.4318182 0.005865885 22.650826 19
[3]  {mushroom cream sauce,pasta}        => {escalope}              0.002532996 0.9500000 0.002666311 11.976387 19
[4]  {parmesan cheese,tomatoes}          => {frozen vegetables}     0.002133049 0.6666667 0.003199573  6.993939 16
[5]  {mineral water,whole wheat pasta}   => {olive oil}             0.003866151 0.4027778 0.009598720  6.115863 29
[6]  {frozen vegetables,parmesan cheese} => {tomatoes}              0.002133049 0.3902439 0.005465938  5.706081 16
[7]  {burgers,herb & pepper}             => {ground beef}           0.002266364 0.5483871 0.004132782  5.581345 17
[8]  {light cream,mineral water}         => {chicken}               0.002399680 0.3272727 0.007332356  5.455273 18
[9]  {ground beef,shrimp}                => {herb & pepper}         0.002932942 0.2558140 0.011465138  5.172131 22
[10] {fromage blanc}                     => {honey}                 0.003332889 0.2450980 0.013598187  5.164271 25
[11] {ground beef,low fat yogurt}        => {herb & pepper}         0.002399680 0.2500000 0.009598720  5.054582 18
[12] {spaghetti,tomato sauce}            => {ground beef}           0.003066258 0.4893617 0.006265831  4.980600 23
[13] {meatballs,spaghetti}               => {tomatoes}              0.002133049 0.3333333 0.006399147  4.873944 16
[14] {light cream}                       => {chicken}               0.004532729 0.2905983 0.015597920  4.843951 34
[15] {frozen vegetables,herb & pepper}   => {ground beef}           0.002799627 0.4666667 0.005999200  4.749616 21
[16] {mineral water,tomato sauce}        => {ground beef}           0.002666311 0.4651163 0.005732569  4.733836 20
[17] {pasta}                             => {escalope}              0.005865885 0.3728814 0.015731236  4.700812 44
[18] {french fries,herb & pepper}        => {ground beef}           0.003199573 0.4615385 0.006932409  4.697422 24
[19] {cereals,spaghetti}                 => {ground beef}           0.003066258 0.4600000 0.006665778  4.681764 23
[20] {french fries,ground beef}          => {herb & pepper}         0.003199573 0.2307692 0.013864818  4.665768 24
[21] {chicken,ground beef}               => {herb & pepper}         0.002133049 0.2253521 0.009465405  4.556243 16
[22] {grated cheese,ground beef}         => {herb & pepper}         0.002532996 0.2235294 0.011331822  4.519391 19
[23] {pasta}                             => {shrimp}                0.005065991 0.3220339 0.015731236  4.506672 38
[24] {chocolate,herb & pepper}           => {ground beef}           0.003999467 0.4411765 0.009065458  4.490183 30
[25] {chicken,herb & pepper}             => {ground beef}           0.002133049 0.4324324 0.004932676  4.401188 16
[26] {cake,frozen vegetables}            => {tomatoes}              0.003066258 0.2987013 0.010265298  4.367560 23
[27] {milk,tomatoes}                     => {soup}                  0.003066258 0.2190476 0.013998134  4.335293 23
[28] {herb & pepper,shrimp}              => {ground beef}           0.002932942 0.4150943 0.007065725  4.224725 22
[29] {eggs,ground beef}                  => {herb & pepper}         0.004132782 0.2066667 0.019997334  4.178455 31
[30] {milk,olive oil}                    => {soup}                  0.003599520 0.2109375 0.017064391  4.174781 27
[31] {herb & pepper,low fat yogurt}      => {ground beef}           0.002399680 0.4090909 0.005865885  4.163624 18
[32] {whole wheat pasta}                 => {olive oil}             0.007998933 0.2714932 0.029462738  4.122410 60
```

Select the rule from Q1 with the greatest lift.

```
#i) which rule has the greatest lift?
inspect(sort(model, by = "lift")[1])
```

///

```
    lhs                                  rhs     support     confidence coverage    lift    count
[1] {escalope,mushroom cream sauce} => {pasta} 0.002532996 0.4418605  0.005732569 28.0881 19
```

///

maximum length of 2.

```
model1 <- apriori(dataset, parameter = list(support =
                                              0.002, confidence = 0.20, maxlen = 2))
```

Select the rule from Q1 with the greatest lift and length of 2:

```
inspect(sort(model1, by = "lift")[1])#2
```

///

```
    lhs                rhs     support     confidence coverage   lift     count
[1] {fromage blanc} => {honey} 0.003332889 0.245098   0.01359819 5.164271 25
```

///

i) Which rule has the better lift?
Rule 1 maxlen equal 3
Rule 1 has the better lift with value = 28.0881.

ii) Which rule has the greater support?
iii) Rule 2 maxlen equal 2
Rule 2 has the greater support with value = 0.003332889.

iv) If you were a marketing manager, and could fund only one of these rules, which would it be, and why?

Rule 1 with maxlen equal 3 as it has better lift and confidence.
Lift gives better correlation measure that judge how many times more often events occur together.
Confidence is a measurement of the predictive power and accuracy. So, the higher the better.
While confidence measures how frequently an itemset occurs the data set and due to the count is higher in rule 2 which will lead to better probability to happen together but not mean that they are reasons for happening together

Part B: Course Recommender System using Collaborative Filtering:

First consider a user-based collaborative filter. This requires computing correlations between all student pairs. For which students is it possible to compute correlations with E.N.? Compute them.

Mean

1) $LN = (4+3+2+4+2)/5 = 3$

2) $MH = (3+4+4)/3 = 3.67$

3) $JH = (2+2)/2 = 2$

4) $EN = (4+4+4+3)/4 = 3.75$

$DU = (4+4)/2 = 4$

$FL = 4$

$GL = 4$

$AH = 3$

$DS = (4+2+4)/3 = 3.33$

EN & LN

$$\text{corr} = \frac{(4-3.75)(4-3)+(4-3.75)(4-3)+(3-3.75)(2-3)}{\sqrt{(4-3.75)^2+(4-3.75)^2+(3-3.75)^2}\sqrt{(4-3)^2+(4-3)^2+(2-3)^2}}$$

$$= 0.870$$

EN & MH

$$\text{corr} = \frac{(4-3.75)(3-3.67)}{\sqrt{(4-3.75)^2}\sqrt{(3-3.67)^2}} = -1$$

EN & JH

$$\text{corr} = \frac{(4-3.75)(2-2)}{\sqrt{(4-3.75)^2}\sqrt{(2-2)^2}} = 0$$

EN & DU

$$corr = \frac{(4-3.75)(4-4)}{\sqrt{(4-3.75)^2}\sqrt{(4-4)^2}} = 0$$

EN & FL = 0

EN & GL = 0

EN & AH = 0

EN & SA = 0

EN & RW = 0

EN & BA = 0

EN & MG = 0

EN & AF = 0

EN & KG = 0

EN & DS

2.5×10^{-3} / 0.7071

$$Corr = \frac{\underbrace{(4-3.75)}_{0.25}\underbrace{(4-3.33)}_{0.67} + \underbrace{(4-3.75)}_{0.25}\underbrace{(2-3.33)}_{-1.33} + \underbrace{(4-3.75)}_{0.25}\underbrace{(4-3.33)}_{0.67}}{\sqrt{\underbrace{(4-3.75)^2}_{0.25} \times 3} * \sqrt{(4-3.33)^2 + (2-3.33)^2 + (4-3.33)^2}}$$

$$= 3.53 \times 10^{-3}$$

2) Based on the single nearest student to E.N., which single course should we recommend to E.N.? Explain why.

E.N and L.N had the highest correlation with value 0.870.

We have 2 choices Python or forecast but we should recommend Python as it had higher ratting.

3) Use R to compute the cosine similarity between users.

```
dataset=read.csv("D:/....... UOTTWA/Fundemental of applied sci/Assignment/4-5/PART2.csv")
rownames(dataset)=dataset$X
dataset = as.matrix(dataset[,-1])
coss=t(dataset)
coss[is.na(coss)]=0
temp=cosine(coss)
```

	LN	MH	JH	EN	DU	FL	GL	AH	SA	RW	BA	MG	AF	KG	DS
LN	1.0000000	0.5354529	0.4040610	0.7190319	0.4040610	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.2020305	0.0000000	0.0000000	0.7619048
MH	0.5354529	1.0000000	0.7730207	0.2482286	0.7730207	0.6246950	0.6246950	0.6246950	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.3123475
JH	0.4040610	0.7730207	1.0000000	0.3746343	1.0000000	0.7071068	0.7071068	0.7071068	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.4714045
EN	0.7190319	0.2482286	0.3746343	1.0000000	0.3746343	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.8830216
DU	0.4040610	0.7730207	1.0000000	0.3746343	1.0000000	0.7071068	0.7071068	0.7071068	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.4714045
FL	0.0000000	0.6246950	0.7071068	0.0000000	0.7071068	1.0000000	1.0000000	1.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
GL	0.0000000	0.6246950	0.7071068	0.0000000	0.7071068	1.0000000	1.0000000	1.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
AH	0.0000000	0.6246950	0.7071068	0.0000000	0.7071068	1.0000000	1.0000000	1.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
SA	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000	0.4472136	1.0000000	0.7071068	1.0000000	1.0000000	0.0000000
RW	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.4472136	1.0000000	0.4472136	0.3162278	0.4472136	0.4472136	0.0000000
BA	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000	0.4472136	1.0000000	0.7071068	1.0000000	1.0000000	0.0000000
MG	0.2020305	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.7071068	0.3162278	0.7071068	1.0000000	0.7071068	0.7071068	0.0000000
AF	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000	0.4472136	1.0000000	0.7071068	1.0000000	1.0000000	0.0000000
KG	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000	0.4472136	1.0000000	0.7071068	1.0000000	1.0000000	0.0000000
DS	0.7619048	0.3123475	0.4714045	0.8830216	0.4714045	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000

4) Based on the cosine similarities of the nearest students to E.N., which course should be recommended to E.N.?

```
#Convert ratings matrix to real rating matrx which makes it dense
datsetmatrix = as(dataset, "realRatingMatrix")
model = Recommender(datsetmatrix, method = "UBCF", param=list(method="Cosine",normalize=NULL,nn=5))
Top_pred = predict(model, datsetmatrix[4], n=1)
#Convert the recommendations to a list
Top_List = as(Top_pred, "list")
Top_List
```

```
$EN
[1] "PYTHON"
```

5) Apply item-based collaborative filtering to this dataset

```
#Apply item-based collaborative filtering to this dataset
model = Recommender(datsetmatrix, method = "IBCF", param=list(method="pearson"))
Top_pred = predict(model, datsetmatrix[4], n=1)
#Convert the recommendations to a list
Top_List = as(Top_pred, "list")
Top_List
```

```
$EN
[1] "SPATIAL"
```