uOttawa

Faculté de génie
Faculty of Engineering

Report Assignment 3

Association Rules and Collaborative Filtering

DT15126 [EG] Fundamentals /Applied Data Sci 20219

Name: Abdallah Medhat Mohamed Rashed
Email: arash015@uOttawa.ca
ID: 300273110

Objectives

The purpose of this assignment to apply Association Rules and Collaborative Filtering on the different data using R and numerical.

Part A: Association Rules numerical

1) Find all frequent itemsets in database X when Using the threshold values support = 25%.
   Calculate support for each item 1st iteration. The color of record is green (deleted).

Item	Count	Support %
A	5	5/8 = 62.5%
B	4	4/8 = 50%
C	5	5/8 = 62.5%
D	6	6/8 = 75%
E	1	1/8 = 12.5%
F	4	4/8 = 50%
G	5	5/8 = 62.5%

Delete item (E) because of support (12.5%) is smaller than threshold of support (25%).

item	Count	Support %
A	5	5/8 = 62.5%
B	4	4/8 = 50%
C	5	5/8 = 62.5%
D	6	6/8 = 75%
F	4	4/8 = 50%
G	5	5/8 = 62.5%

Calculate support for each items 2nd iteration.

Items	Count	Support %
A,B	3	3/8 = 37.5%
A,C	3	3/8 = 37.5%
A,D	4	4/8 = 50%
A,F	2	2/8 = 25%
A,G	2	2/8 = 25%
B,C	2	2/8 = 25%
B,D	2	2/8 = 25%
B,F	1	1/8 = 12.5%
B,G	2	2/8 = 25%
C,D	4	4/8 = 50%
C,F	2	2/8 = 25%
C,G	3	3/8 = 37.5%
D,F	4	4/8 = 50%
D,G	3	3/8 = 37.5%
F,G	2	2/8 = 25%

Delete item (B, F) because of support (12.5%) is smaller than threshold of support (25%).

Items	Count	Support %
A,B	3	3/8 = 37.5%
A,C	3	3/8 = 37.5%
A,D	4	4/8 = 50%
A,F	2	2/8 = 25%
A,G	2	2/8 = 25%
B,C	2	2/8 = 25%
B,D	2	2/8 = 25%
B,G	2	2/8 = 25%
C,D	4	4/8 = 50%
C,F	2	2/8 = 25%
C,G	3	3/8 = 37.5%
D,F	4	4/8 = 50%
D,G	3	3/8 = 37.5%
F,G	2	2/8 = 25%

Calculate support for each items 3rd iteration.

Items	Count	Support %
A,B,C	1	1/8=12.5%
A,B,D	2	2/8=25%
A,B,G	1	1/8=12.5%
A,C,D	3	3/8=37.5%
A,C,F	1	1/8=12.5%
A,C,G	1	1/8=12.5%
A,D,F	2	2/8=25%
A,D,G	1	1/8=12.5%
A,F,G	0	0/8=0%
B,C,D	1	1/8=12.5%
B,C,G	1	1/8=12.5%
B,D,G	0	0/8=0%
C,D,F	2	2/8=25%
C,D,G	2	1/8=12.5%
C,F,G	1	1/8=12.5%
D,F,G	2	2/8=25%

Delete items (A,B,C),(A,B,G),(A,C,F),(A,C,G),(A,D,G),(A,F,G),(B,C,D),(B,C,G),(B,D,G),(C,F,G) because of support (12.5%)(0%) is smaller than threshold of support (25%).

Items	Count	Support %
A,B,D	2	2/8=25%
A,C,D	3	3/8=37.5%
A,D,F	2	2/8=25%
C,D,F	2	2/8=25%
C,D,G	2	1/8=12.5%
D,F,G	2	2/8=25%

2) Find strong association rules for database X when using Confidence 60 % (threshold). The color of record is red (Confidence value greater than or equal Confidence 60 %).

Item set	Confidence	State [conf > =threshold]
{A, B} -> D	2/3 = 0.67	True
{A, D} -> B	2/4 = 0.5	False
{B, D} -> A	2/2 = 1	True
A -> {B, D}	2/5 = 0.4	False
B -> {A, D}	2/4 = 0.5	False
D -> {B, A}	2/6 = 0.33	False

Item set	Confidence	State [conf > =threshold]
{A, C} -> D	3/3 = 1	True
{A, D} -> C	3/4 = 0.75	True
{C, D} -> A	3/4 = 0.75	True
A -> {C, D}	3/5 = 0.6	True
C -> {A, D}	3/5 = 0.6	True
D -> {C, A}	3/6 = 0.5	False

Item set	Confidence	State [conf > =threshold]
{A, F} -> D	2/2 = 1	True
{A, D} -> F	2/4 = 0.5	False
{F, D} -> A	2/4 = 0.5	False
A -> {F, D}	2/5 = 0.4	False
F -> {A, D}	2/4 = 0.5	False

D -> {F, A}	2/6 = 0.33	False
Item set	Confidence	State [conf > =threshold]
{C, F} -> D	2/2 = 1	True
{C, D} -> F	2/4 = 0.5	False
{F, D} -> C	2/4 = 0.5	False
F -> {C, D}	2/4 = 0.5	False
C -> {D, F}	2/5 = 0.4	False
D -> {C, F}	2/6 = 0.33	False

Item set	Confidence	State [conf > =threshold]
{G, F} -> D	2/2 = 1	True
{F, D} -> G	2/4 = 0.5	False
{G, D} -> F	2/3 = 0.67	True
G -> {F, D}	2/5 = 0.4	False
F -> {G, D}	2/4 = 0.5	False
D -> {G, F}	2/6 = 0.33	False

Item set	Confidence	State [conf > =threshold]
{G, C} -> D	2/3 = 0.67	True
{C, D} -> G	2/4 = 0.5	False
{G, D} -> C	2/3 = 0.67	True
G -> {C, D}	2/5 = 0.4	False
C -> {G, D}	2/5 = 0.4	False
D -> {G, C}	2/6 = 0.33	False

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

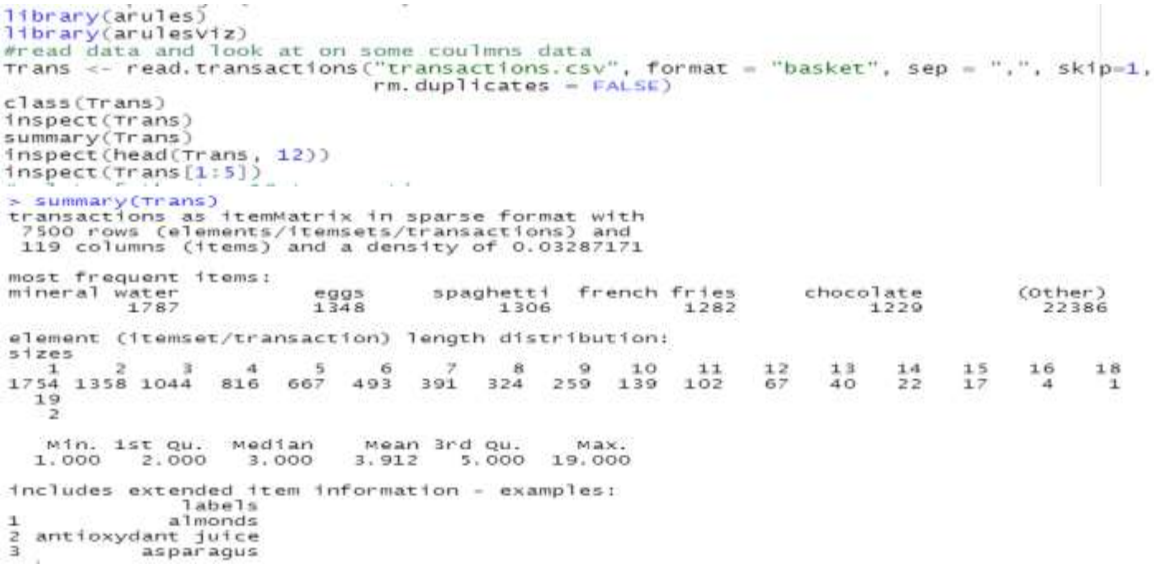
The color of record is red (when lift is less than 1, the correlate is negative).

Rule	Lift	State
{A, B} -> D	0.25 / (0.375 * 0.75) = 0.89	negatively correlated
{B, D} -> A	0.25 / (0.25 * 0.625) = 1.6	positively correlated
{A, C} -> D	0.375 / (0.375 * 0.75) = 1.3	positively correlated
{A, D} -> C	0.375 / (0.5 * 0.625) = 1.2	positively correlated
{C, D} -> A	0.375 / (0.5 * 0.625) = 1.2	positively correlated
A -> {C, D}	0.375 / (0.5 * 0.625) = 1.2	positively correlated
C -> {A, D}	0.375 / (0.5 * 0.625) = 1.2	positively correlated
{A, F} -> D	0.25 / (0.25 * 0.75) = 1.3	positively correlated
{C, F} -> D	0.25 / (0.25 * 0.75) = 1.3	positively correlated
{G, F} -> D	0.25 / (0.25 * 0.75) = 1.3	positively correlated
{G, D} -> F	0.25 / (0.375 * 0.5) = 1.3	positively correlated

{G, C} -> D	0.25 / (0.375 * 0.75) = 0.89	negatively correlated
{G, D} -> C	0.25/ (0.375 * 0.625) = 1.067	positively correlated

Association Rules by R:

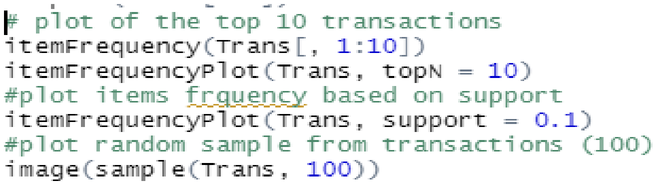
1) Read transactions data (transactions.csv) and get some information about transactions data.



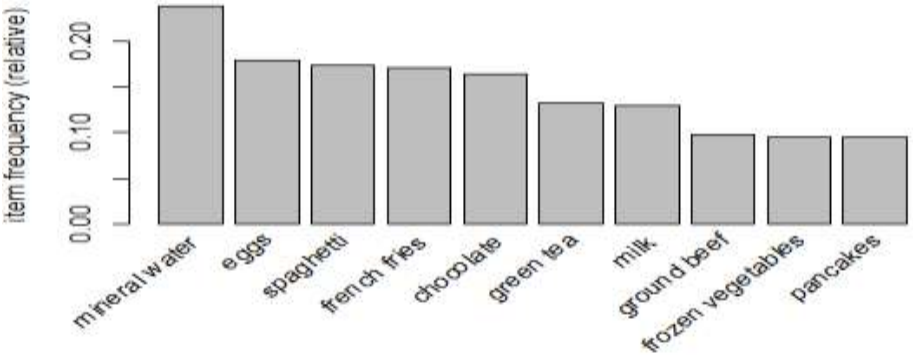
```
library(arules)
library(arulesviz)
#read data and look at on some coulmns data
Trans <- read.transactions("transactions.csv", format = "basket", sep = ",", skip=1,
                           rm.duplicates = FALSE)
class(Trans)
inspect(Trans)
summary(Trans)
inspect(head(Trans, 12))
inspect(Trans[1:5])
```

```
> summary(Trans)
transactions as itemMatrix in sparse format with
 7500 rows (elements/itemsets/transactions) and
 119 columns (items) and a density of 0.03287171

most frequent items:
mineral water          eggs     spaghetti  french fries     chocolate         (Other)
         1787          1348          1306          1282          1229           22386

element (itemset/transaction) length distribution:
sizes
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   18
1754 1358 1044  816  667  493  391  324  259  139  102   67   40   22   17    4    1
  19
   2

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   3.912   5.000  19.000

includes extended item information - examples:
            labels
1          almonds
2 antioxydant juice
3        asparagus
```

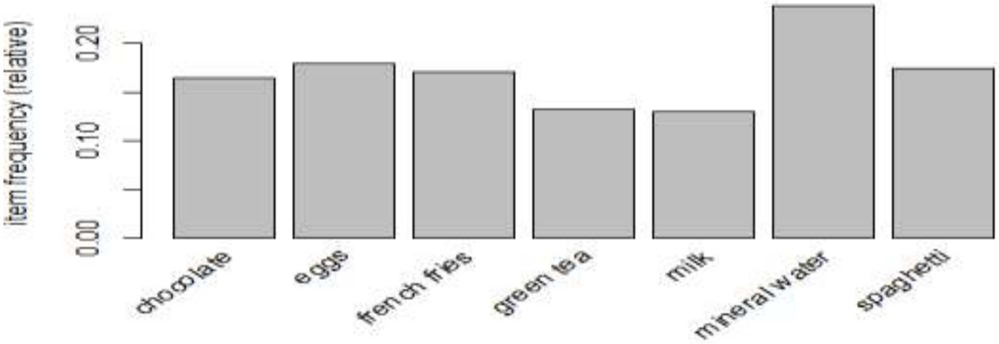
2) Generate a plot of the top 10 transactions



```
# plot of the top 10 transactions
itemFrequency(Trans[, 1:10])
itemFrequencyPlot(Trans, topN = 10)
#plot items frquency based on support
itemFrequencyPlot(Trans, support = 0.1)
#plot random sample from transactions (100)
image(sample(Trans, 100))
```

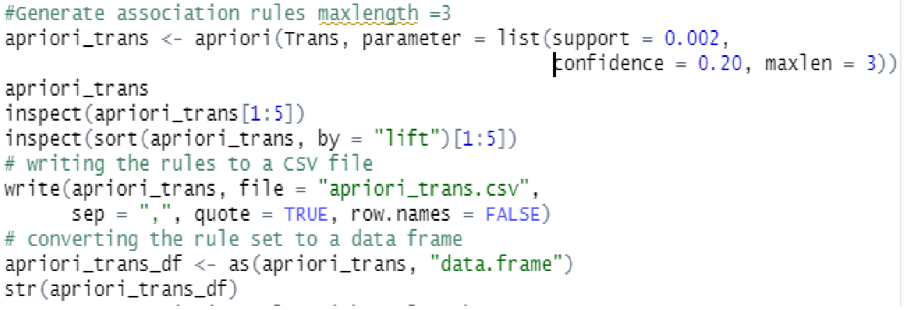


The plot of the top 10 transactions.

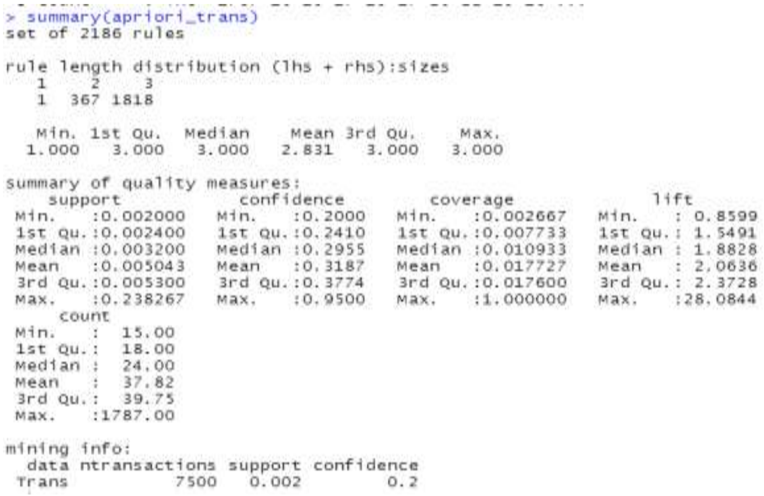


The plot transactions based on support =0.1

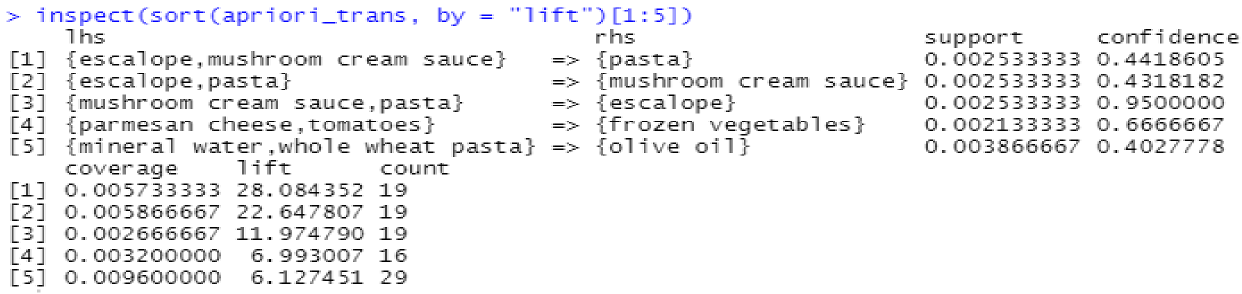
4) Generate association rules using minimum support of 0.002, minimum confidence of 0.20, and maximum length of 3 and save the rule in csv file.



```
#Generate association rules maxlength =3
apriori_trans <- apriori(Trans, parameter = list(support = 0.002,
                                                 confidence = 0.20, maxlen = 3))
apriori_trans
inspect(apriori_trans[1:5])
inspect(sort(apriori_trans, by = "lift")[1:5])
# writing the rules to a csv file
write(apriori_trans, file = "apriori_trans.csv",
      sep = ",", quote = TRUE, row.names = FALSE)
# converting the rule set to a data frame
apriori_trans_df <- as(apriori_trans, "data.frame")
str(apriori_trans_df)
```

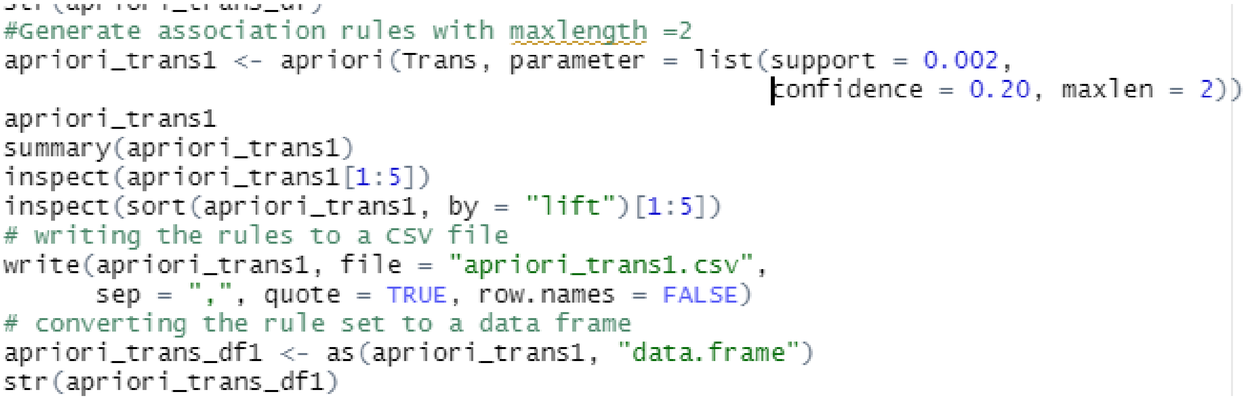


```
> summary(apriori_trans)
set of 2186 rules

rule length distribution (lhs + rhs):sizes
   1    2    3 
   1  367 1818 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   3.000   3.000   2.831   3.000   3.000 

summary of quality measures:
    support           confidence        coverage             lift        
 Min.   :0.002000   Min.   :0.2000   Min.   :0.002667   Min.   : 0.8599  
 1st Qu.:0.002400   1st Qu.:0.2410   1st Qu.:0.007733   1st Qu.: 1.5491  
 Median :0.003200   Median :0.2955   Median :0.010933   Median : 1.8828  
 Mean   :0.005043   Mean   :0.3187   Mean   :0.017727   Mean   : 2.0636  
 3rd Qu.:0.005300   3rd Qu.:0.3774   3rd Qu.:0.017600   3rd Qu.: 2.3728  
 Max.   :0.238267   Max.   :0.9500   Max.   :1.000000   Max.   :28.0844  
     count        
 Min.   :  15.00  
 1st Qu.:  18.00  
 Median :  24.00  
 Mean   :  37.82  
 3rd Qu.:  39.75  
 Max.   :1787.00  

mining info:
  data ntransactions support confidence
 Trans          7500   0.002        0.2
```



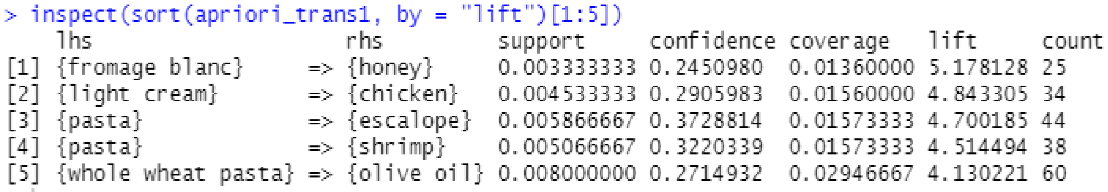
```
> inspect(sort(apriori_trans, by = "lift")[1:5])
    lhs                                     rhs                    support     confidence
[1] {escalope,mushroom cream sauce}      => {pasta}                0.002533333 0.4418605 
[2] {escalope,pasta}                     => {mushroom cream sauce} 0.002533333 0.4318182 
[3] {mushroom cream sauce,pasta}         => {escalope}             0.002533333 0.9500000 
[4] {parmesan cheese,tomatoes}           => {frozen vegetables}    0.002133333 0.6666667 
[5] {mineral water,whole wheat pasta}    => {olive oil}            0.003866667 0.4027778 
    coverage    lift      count
[1] 0.005733333 28.084352 19   
[2] 0.005866667 22.647807 19   
[3] 0.002666667 11.974790 19   
[4] 0.003200000  6.993007 16   
[5] 0.009600000  6.127451 29   
```

5) Generate association rules using minimum support of 0.002, minimum confidence of 0.20, and maximum length of 2 and save the rule in csv file.



```
#Generate association rules with maxlength =2
apriori_trans1 <- apriori(Trans, parameter = list(support = 0.002,
                                                  confidence = 0.20, maxlen = 2))
apriori_trans1
summary(apriori_trans1)
inspect(apriori_trans1[1:5])
inspect(sort(apriori_trans1, by = "lift")[1:5])
# writing the rules to a csv file
write(apriori_trans1, file = "apriori_trans1.csv",
      sep = ",", quote = TRUE, row.names = FALSE)
# converting the rule set to a data frame
apriori_trans_df1 <- as(apriori_trans1, "data.frame")
str(apriori_trans_df1)
```

```
> summary(apriori_trans1)
set of 368 rules

rule length distribution (lhs + rhs):sizes
  1   2 
  1 367 

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   2.000   2.000   1.997   2.000   2.000 

summary of quality measures:
    support           confidence        coverage             lift       
 Min.   :0.002000   Min.   :0.2000   Min.   :0.00480   Min.   :0.8599  
 1st Qu.:0.003333   1st Qu.:0.2244   1st Qu.:0.01187   1st Qu.:1.3123  
 Median :0.006067   Median :0.2519   Median :0.02253   Median :1.5352  
 Mean   :0.011114   Mean   :0.2748   Mean   :0.04295   Mean   :1.6871  
 3rd Qu.:0.014433   3rd Qu.:0.3158   3rd Qu.:0.05240   3rd Qu.:1.8438  
 Max.   :0.238267   Max.   :0.4872   Max.   :1.00000   Max.   :5.1781  
     count        
 Min.   :  15.00  
 1st Qu.:  25.00  
 Median :  45.50  
 Mean   :  83.35  
 3rd Qu.: 108.25  
 Max.   :1787.00  

mining info:
  data ntransactions support confidence
 Trans          7500   0.002        0.2
```



```
> inspect(sort(apriori_trans1, by = "lift")[1:5])
    lhs                   rhs          support     confidence coverage   lift     count
[1] {fromage blanc}     => {honey}     0.003333333 0.2450980  0.01360000 5.178128 25
[2] {light cream}       => {chicken}   0.004533333 0.2905983  0.01560000 4.843305 34
[3] {pasta}             => {escalope}  0.005866667 0.3728814  0.01573333 4.700185 44
[4] {pasta}             => {shrimp}    0.005066667 0.3220339  0.01573333 4.514494 38
[5] {whole wheat pasta} => {olive oil} 0.008000000 0.2714932  0.02946667 4.130221 60
```

The highest association rules of lift when Maximum length = 3 and Maximum length=2.

Maximum length	Rules	Support	Confidence	Lift	count
3	{Escalope, mushroom cream sauce} =>{pasta}	0.002533333	0.4418605	28.084352	19
2	{fromage blanc} => {honey}	0.003333333	0.2450980	5.178128	25

Which rule has the better lift?

Rules with Maximum length 3 that {Escalope, mushroom cream sauce} => {pasta} has better lift with value 28.084352.

Which rule has the greater support?

Rules with Maximum length 2 that {fromage blanc} => {honey} has greater Support with value 0.00333333.

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

I will choose Rules that with Maximum length 3 that {Escalope, mushroom cream sauce} => {pasta}, because it has highest Values in confidence, Lift and coverage. If I apply this rules in sorting market item it will get more money and profit.

Part B: Collaborative Filtering numerical

1) For which students is it possible to compute correlations with E.N? By calculate average for all students and calculate correlation Between EN Student and others.

   $Average_i=x_1+x_2+x_3+………/n$

	SQL	Spatial	PA1	DM in R	Python	forecast	R Prog	Hadoop	Regression	Average
LN	4				3	2	4		2	3
MH	3	4			4					3.666667
JH	2	2								2
EN	4			4			4		3	3.75
DU	4	4								4
FL		4								4
GL		4								4
AH		3								3
SA			4							4
RW			2					4		3
BA			4							4
MG			4			4				4
AF			4							4
KG			3							3
DS	4			2			4			3.333333

$Corr(U_1,U_2)= \sum(r_{1,i}-r_1\ (average))(r_{2,i}-r_2(average))/\ ((\ \sum(r_{1,i}-r_1\ (average))\ ^2)^{1/2}((r_{2,i}-r_2(average))^2\ )^{1/2}$

Correlated student	Equation	value
Corr(EN ,LN)	$= (4-3.75)(4-3)+(4-3.75)(4-3)+(3-3.75)(2-3)\ /$ $((4-3.75)^2+(4-3.75)^2+(3-3.75)^2)^{1/2}((4-3)^2+(4-3)^2+(2-3)^2)^{1/2}$	= 0.87
Corr(EN ,MH)	$((4-3.75)(3-3.67)\ /\ ((4-3.75)^2)^{1/2}\ ((3-3.67)^2)^{1/2}$	= -1
Corr(EN ,JH)	$((4-3.75)(2-2)\ /\ ((4-3.75)^2)^{1/2}\ ((2-2)^2)^{1/2}$	= 0
Corr(EN ,DU)	$((4-3.75)(4-4)\ /\ ((4-3.75)^2)^{1/2}\ ((4-4)^2)^{1/2}$	= 0
Corr(EN ,DS)	$(4-3.75)(4-3.33)+(4-3.75)(2-3.33)+(4-3.75)(4-3.33)/($ $(4-3.75)^2+(4-3.75)^2+(4-3.75)^2)^{1/2}((4-3.33)^2+(2-3.33)^2+(4-3.33)^2)^{1/2}$	0.003535

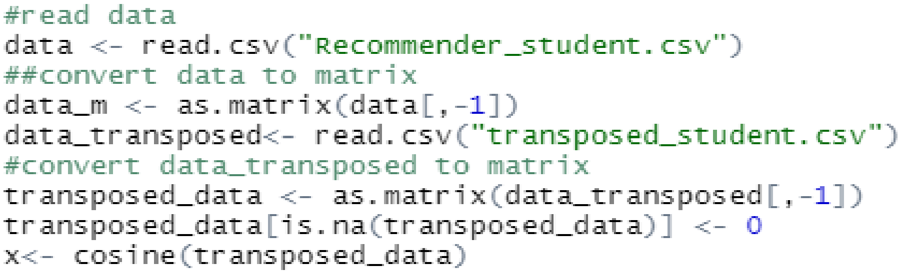
L.N. Student is the highest correlations with E.N. student.

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

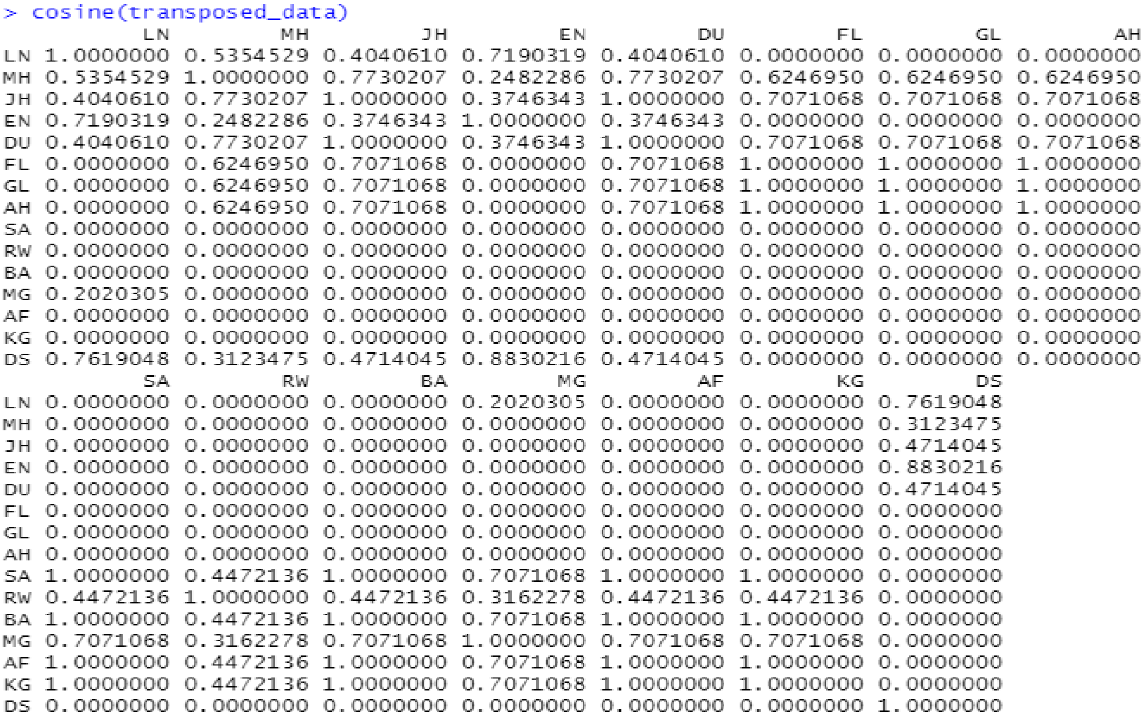
After calculate corr between the students, we obtain the highest correlated between E.N and LN. Dropping the smaller courses between them and remain two course Python and forecast I choice Python Course because python course has highest Rate (3).

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

Creating CSV file has our dataset and other has transpose of it.

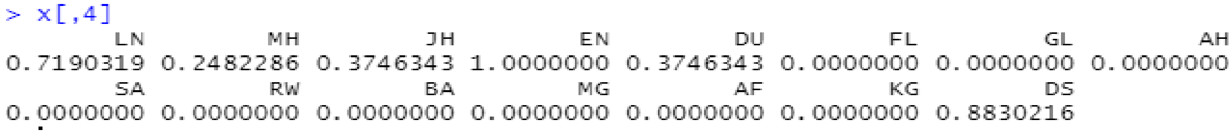


```
#read data
data <- read.csv("Recommender_student.csv")
##convert data to matrix
data_m <- as.matrix(data[,-1])
data_transposed<- read.csv("transposed_student.csv")
#convert data_transposed to matrix
transposed_data <- as.matrix(data_transposed[,-1])
transposed_data[is.na(transposed_data)] <- 0
x<- cosine(transposed_data)
```



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

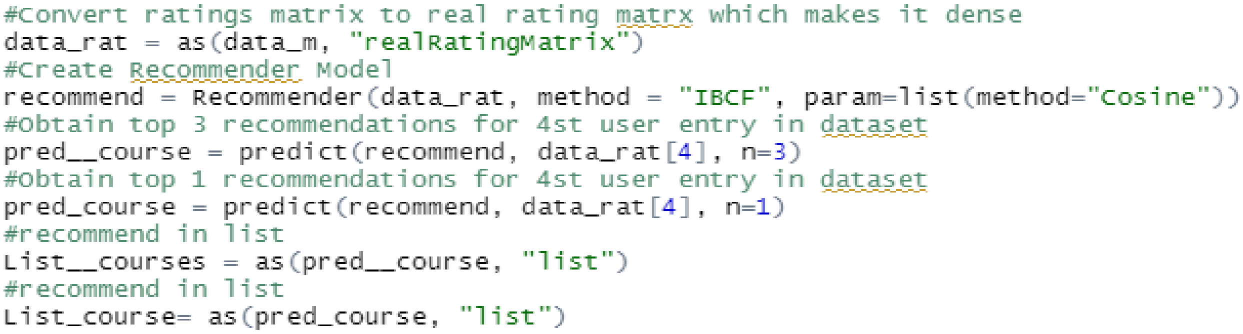
4) Based on the cosine similarities of the nearest students to E.N., which course should be recommended to E.N.?



```
> x[,4]
       LN        MH        JH        EN        DU        FL        GL        AH 
0.7190319 0.2482286 0.3746343 1.0000000 0.3746343 0.0000000 0.0000000 0.0000000 
       SA        RW        BA        MG        AF        KG        DS 
0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.8830216 
```

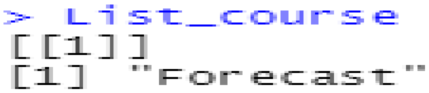
We obtain D.S. and L.N. students are the highest correlated D.S. has no new courses to recommend to E.N based on cosine similarities. EN has two choice python and Forecast based on the correlated with LN we choice python course to recommend to E.N. student based on rating.

5) Apply item-based collaborative filtering to this dataset (using R) and based on the results, recommend a course to E.N



```
#Convert ratings matrix to real rating matrx which makes it dense
data_rat = as(data_m, "realRatingMatrix")
#Create Recommender Model
recommend = Recommender(data_rat, method = "IBCF", param=list(method="Cosine"))
#Obtain top 3 recommendations for 4st user entry in dataset
pred__course = predict(recommend, data_rat[4], n=3)
#Obtain top 1 recommendations for 4st user entry in dataset
pred_course = predict(recommend, data_rat[4], n=1)
#recommend in list
List__courses = as(pred__course, "list")
#recommend in list
List_course= as(pred_course, "list")
```

```
> List__courses
[[1]]
[1] "Forecast" "Spatial"  "Python"
```



```
> List_course
[[1]]
[1] "Forecast"
```

We obtain recommend 3 courses (forecast, spatial and python) to E.N. Student based on item-based collaborative filtering using Recommender function.

We obtain recommend one course (forecast) to E.N. Student based on item-based collaborative filtering using Recommender function.