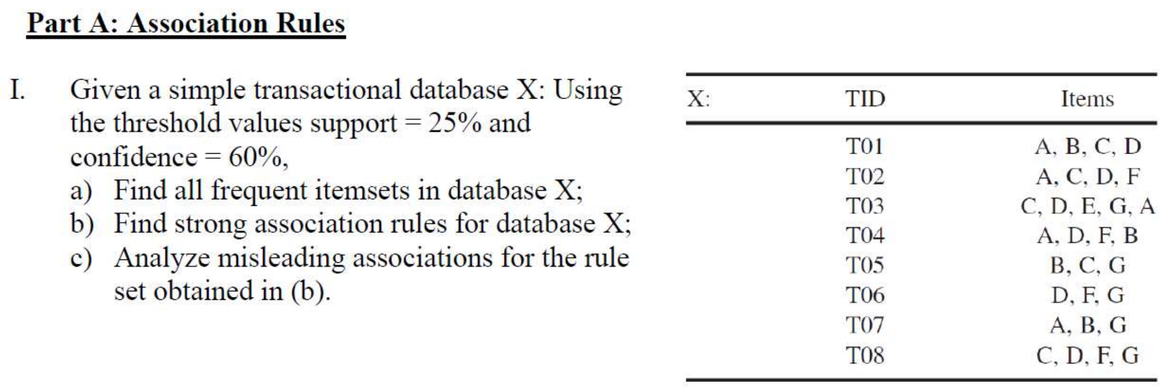
DTI 5126[EG]: Fundamentals for Applied Data Science

Summer 2021
Assignment 3 Association rules and collaborative filters Using R

By
Omar Sorour

Part A



Part A: Association Rules

I. Given a simple transactional database X: Using the threshold values support = 25% and confidence = 60%,

a) Find all frequent itemsets in database X;

b) Find strong association rules for database X;

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

X:	TID	Items
	T01	A, B, C, D
	T02	A, C, D, F
	T03	C, D, E, G, A
	T04	A, D, F, B
	T05	B, C, G
	T06	D, F, G
	T07	A, B, G
	T08	C, D, F, G

a) First, I created a table that has the same dataset but in a more representable way as the following:

trans	A	B	C	D	E	F	G	Column1
tr1	1	1	1	1	0	0	0	ABCD
tr2	1	0	1	1	0	1	0	ACDF
tr3	1	0	1	1	1	0	1	CDEGA
tr4	1	1	0	1	0	1	0	ADFB
tr5	0	1	1	0	0	0	1	BCG
tr6	0	0	0	1	0	1	1	DFG
tr7	1	1	0	0	0	0	1	ABG
tr8	0	0	1	1	0	1	1	CDFG

Then we calculated the frequency table of each item:

1-Item set	Frequency
A	5
B	4
C	5
D	6
E	1
F	4
G	5

The support in our case is 25% which is 2/8 so, all frequencies less than 2 we be discarded (i.e. E item will be dropped)

Here is the corresponding table:

trans	A	B	C	D	F	G	Column1
tr1	1	1	1	1	0	0	ABCD
tr2	1	0	1	1	1	0	ACDF
tr3	1	0	1	1	0	1	CDEGA
tr4	1	1	0	1	1	0	ADFB
tr5	0	1	1	0	0	1	BCG
tr6	0	0	0	1	1	1	DFG
tr7	1	1	0	0	0	1	ABG
tr8	0	0	1	1	1	1	CDFG

The second iteration was for 2-item set

2-Item set	Frequency
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

Only the combination B,F will be removed

2-Item set	Frequency
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

Also the third iteration was for 3-item set

3-Item set	Frequency
A,B,C	1
A,B,D	2
A,B,F	1
A,B,G	1
A,C,D	3
A,C,F	1
A,C,G	1
A,D,F	2
A,D,G	1
A,F,G	0
B,C,D	1
B,C,F	0
B,C,G	1
B,D,F	1
B,D,G	0
B,F,G	0
C,D,f	2
C,D,G	2
C,F,G	1
D,F,G	2

I selected only combinations that had frequency equal or more then 2

3-Item set	Frequency
A,B,D	2
A,C,D	3
A,D,F	2
C,D,f	2
C,D,G	2
D,F,G	2

The fourth iteration for 4-items

4-Item set	Frequency
A, B, C, D	1
A, B, C, F	0
A, B, C, G	0
A, B, D, F	1
A, B, D, G	0
A, B, F, G	0
A, C, D, F	1
A, C, D, G	1
A, C, F, G	0
A, D, F, G	0
B, C, D, F	0
B, C, D, G	0
B, C, F, G	0
B, D, F, G	0
C, D, F, G	1

All under the support as we can see.

b) Finding the strong association rules

After this I started writing the association rules for which Confidence C is equal or more than 60% as the following:

1. {A} => {B}, C= $\frac{s(A,B)}{S(A)} = $ 3/5 = 60%
2. {A}=>{C}, C = 3/5= 60%
3. {A}=> {D}, C = 4/5 =80%
4. {B}=> {A}, C = 3/4 = 75%
5. {C}=>{A}, C = 3/5 = 60%
6. {C}=>{D}, C = 4/5 = 80%
7. {C}=>{G} , C = 3/5 = 60%
8. {D}=>{A}, C = 4/6 = 66.6%
9. {D}=>{C}, C = 4/6 = 66.6%
10. {D}=>{F}, C = 4/6 = 66.6%
11. {F} => {D}, C = 4/4 = 100%
12. {G}=> {C}, C = 3/5 = 60%
13. {G} => {D}, C =3/5 = 60%
14. {A,B}=>{D}, C = 2/3 =66.6%
15. {A,C) => {D}, C = 3/3 = 100%
16. {A,D}=> {C} , C = 3/4 = 75%
17. {A,F} =>{D} , C = 2/2 = 100%
18. {B,D} => {A}, C = 2/2 = 100%
19. {C,D}=>{A} , C =3/4 = 75%
20. {C,F}=>{D} , C = 2/2 = 75%

21. {C,G}=> {D}, C =2/3 =66.6%
22. {C,G}=>{F} , C = 2/3 = 66.6%
23. {D,G} => {F}, C =2/3 =66.6%
24. {F,G} => {D}, C = 2/2 = 100%

c) Analyzing the misleading association rules:
Not all the discovered strong association rules (i.e., passing the required support *s* and required confidence *c*) are interesting enough to be presented and used.

To filter out such misleading associations, one may define that an association rule *A B* is *interesting* if its confidence exceeds a certain measure. The simple argument we used in the example above suggests that the right heuristic to measure association should be

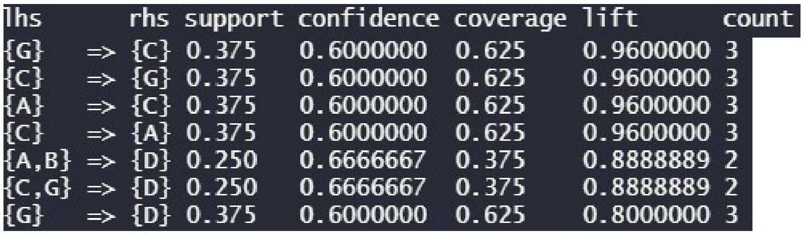
$$\frac{S(A,B)}{s(A)} - s(B) > d$$

If the left-hand side of this inequality equation is negative then we can say that this association rule is not only not interesting but also mis-leading.

Applying the previous equation to the previous association rules generated in b resulted in:

1. {A}=>{C}, C = 3/5= 60% Value = -2.5%
2. {C}=>{A}, C = 3/5 = 60% Value = -2.5%
3. {C}=>{G}, C = 3/5 = 60% Value = -2.5%
4. {G}=> {C}, C = 3/5 = 60% Value = -2.5%
5. {G} => {D}, C =3/5 = 60% Value = -15%
6. {A,B}=>{D}, C = 2/3 =66.6% Value = -8.4%
7. {C,G}=> {D}, C =2/3 =66.6% Value = -8.4%

I also made sure of these misleading association rules using another approach which is the lift index Using R, where value of the lift is less than 1



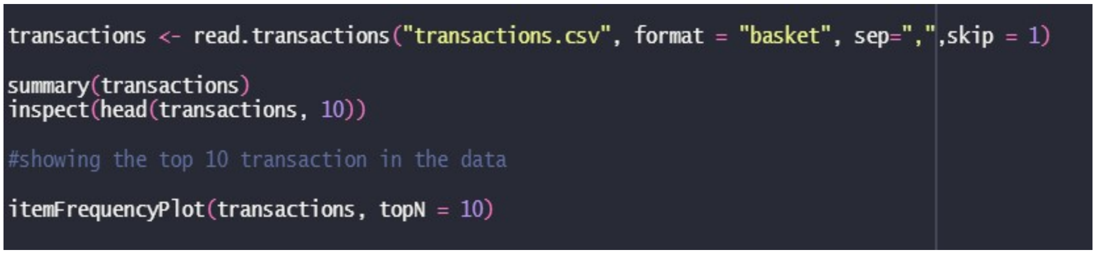
```
lhs       rhs support confidence coverage lift      count
{G}    => {C} 0.375   0.6000000  0.625    0.9600000 3
{C}    => {G} 0.375   0.6000000  0.625    0.9600000 3
{A}    => {C} 0.375   0.6000000  0.625    0.9600000 3
{C}    => {A} 0.375   0.6000000  0.625    0.9600000 3
{A,B} => {D} 0.250   0.6666667  0.375    0.8888889 2
{C,G} => {D} 0.250   0.6666667  0.375    0.8888889 2
{G}    => {D} 0.375   0.6000000  0.625    0.8000000 3
```

Taking the first rule {A}=>{C} as an example for analyzing these mis-leading association rules the percent of the customer who bought item A is 62.5% and who bought B is also 62.5% and who bought both A and C is 37.5% among those 37.5% only 60% who bought A also bought C which is less than the original percent of the customers who bought only C (62.5%) this means that buy item A is not necessarily entails buying item C, that’s why this rule is misleading.

II. A store is interested in determining the associations between items purchased from its Departments. The store chose to conduct a market basket analysis of specific items purchased to analyze customer's buying behavior. You are hereby provided with a file '*transactions.csv*' containing information for transactions made over the past 3 months.

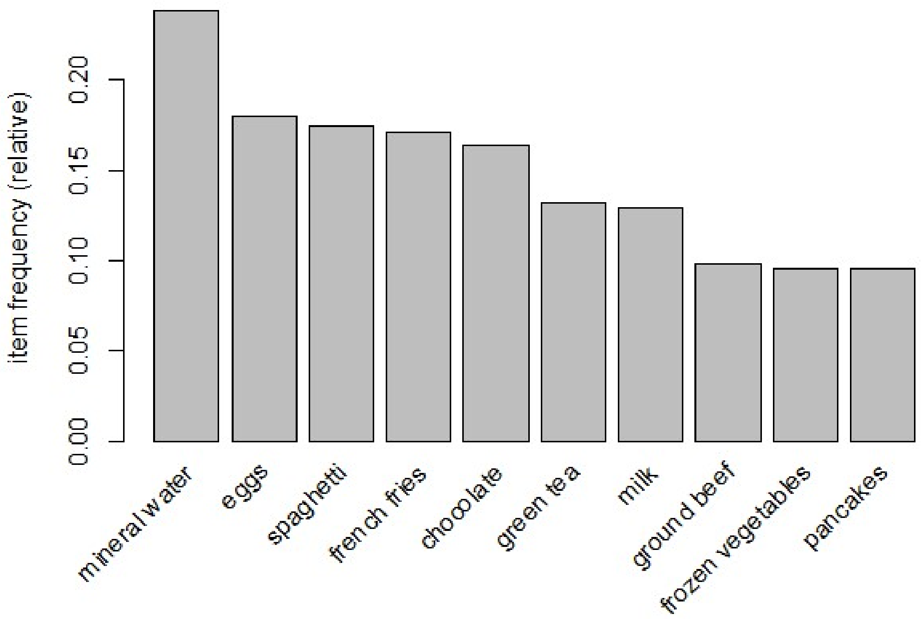
a) Generate a plot of the top 10 transactions
b) Generate association rules using minimum support of 0.002, minimum confidence of 0.20, and maximum length of 3. Display the rules, sorted by descending lift value.
c) Select the rule from Q1 with the greatest lift. Compare this rule with the highest lift rule for maximum length of 2.
   i) Which rule has the better lift?
   ii) Which rule has the greater support?
   iii) If you were a marketing manager, and could fund only one of these rules, which would it be, and why?

a) This was implemented in the following lines of code:

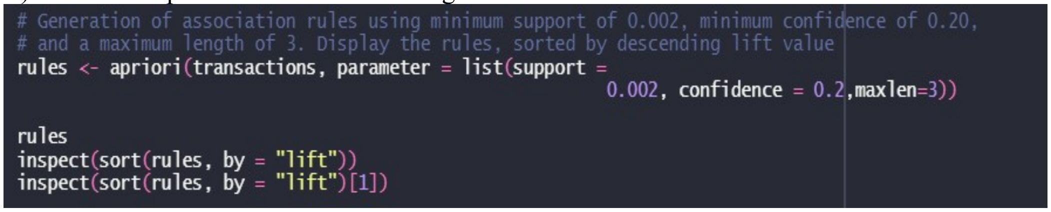


```
transactions <- read.transactions("transactions.csv", format = "basket", sep=",",skip = 1)

summary(transactions)
inspect(head(transactions, 10))

#showing the top 10 transaction in the data

itemFrequencyPlot(transactions, topN = 10)
```

The results:

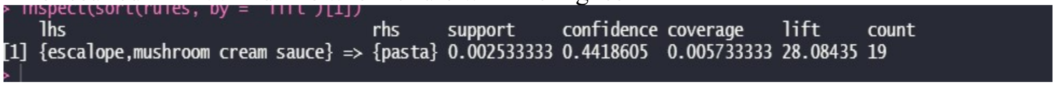


B) This was implemented in the following lines of code:



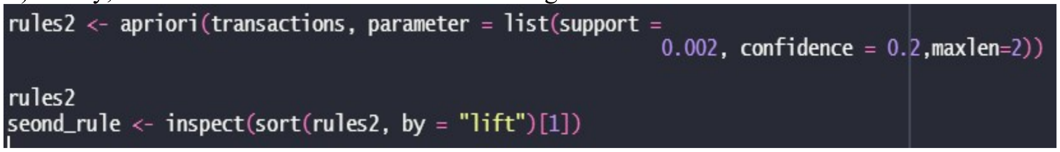
```
# Generation of association rules using minimum support of 0.002, minimum confidence of 0.20,
# and a maximum length of 3. Display the rules, sorted by descending lift value
rules <- apriori(transactions, parameter = list(support =
                                                 0.002, confidence = 0.2,maxlen=3))

rules
inspect(sort(rules, by = "lift"))
inspect(sort(rules, by = "lift")[1])
```

This resulted in 2186 and this was the rule with the highest lift



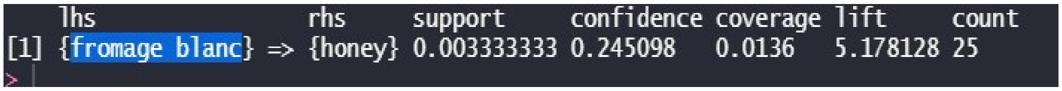
```
> inspect(sort(rules, by = "lift")[1])
    lhs                                rhs     support     confidence coverage    lift     count
[1] {escalope,mushroom cream sauce} => {pasta} 0.002533333 0.4418605  0.005733333 28.08435 19
>
```

C) Firstly, I created set of rules with maximum length = 2



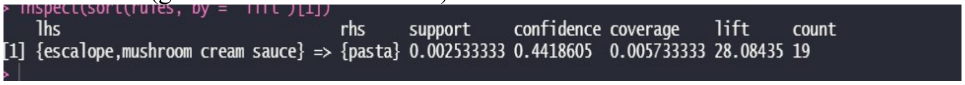
```
rules2 <- apriori(transactions, parameter = list(support =
                                                  0.002, confidence = 0.2,maxlen=2))

rules2
seond_rule <- inspect(sort(rules2, by = "lift")[1])
```

The resulted rule was:



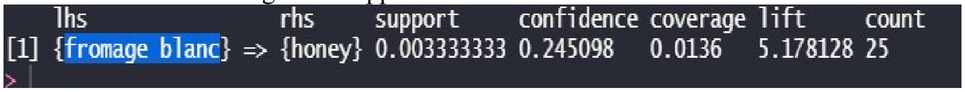
```
    lhs               rhs     support     confidence coverage lift     count
[1] {fromage blanc} => {honey} 0.003333333 0.245098   0.0136   5.178128 25
>
```

i) The first rule (generated when maxlen=3) has the better lift.



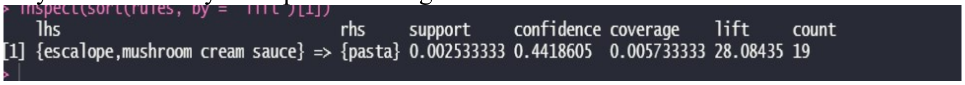
```
> inspect(sort(rules, by = "lift")[1])
    lhs                                rhs     support     confidence coverage    lift     count
[1] {escalope,mushroom cream sauce} => {pasta} 0.002533333 0.4418605  0.005733333 28.08435 19
>
```

ii) The second rule has the greater support.

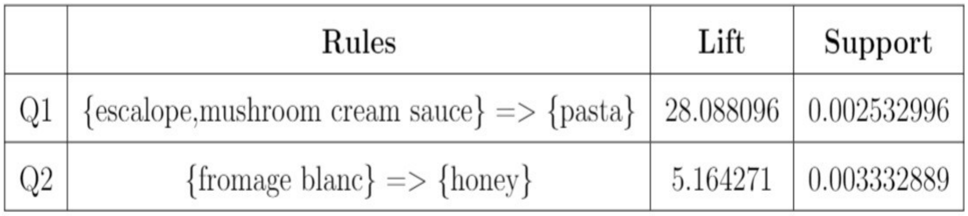


```
    lhs               rhs     support     confidence coverage lift     count
[1] {fromage blanc} => {honey} 0.003333333 0.245098   0.0136   5.178128 25
>
```

iii) I would go for the first rule as it makes more sense as the items are ingredients of lunch meal and they are more likely to be purchased together.



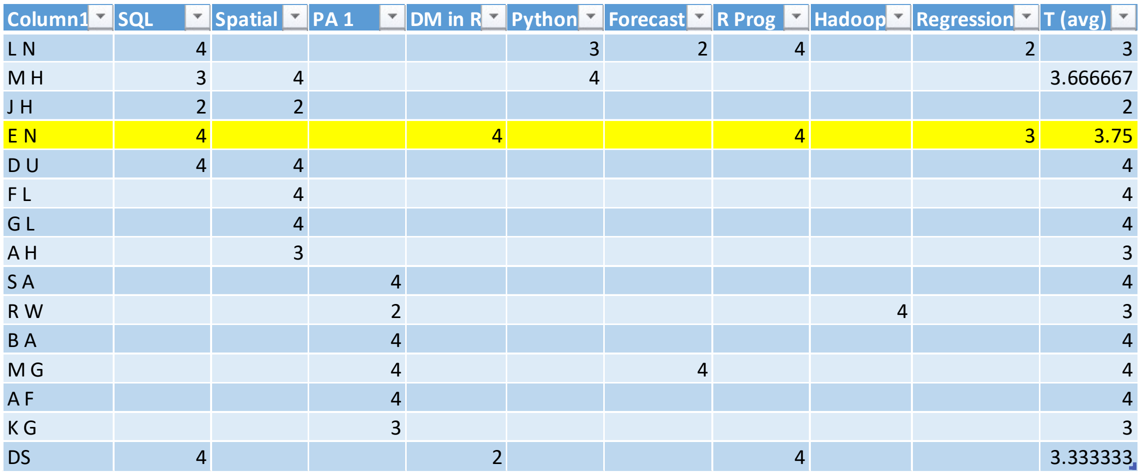
```
> inspect(sort(rules, by = "lift")[1])
    lhs                                rhs     support     confidence coverage    lift     count
[1] {escalope,mushroom cream sauce} => {pasta} 0.002533333 0.4418605  0.005733333 28.08435 19
>
```



	Rules	Lift	Support
Q1	{escalope,mushroom cream sauce} => {pasta}	28.088096	0.002532996
Q2	{fromage blanc} => {honey}	5.164271	0.003332889

1) **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.**

I first created the table in the excel and another column to the dataset that holds the average rating values for each user.



Column1	SQL	Spatial	PA 1	DM in R	Python	Forecast	R Prog	Hadoop	Regression	T (avg)
L N	4				3	2	4		2	3
M H	3	4			4					3.666667
J H	2	2								2
E N	4			4			4		3	3.75
D U	4	4								4
F L		4								4
G L		4								4
A H		3								3
S A			4							4
R W			2					4		3
B A			4							4
M G			4			4				4
A F			4							4
K G			3							3
DS	4			2			4			3.333333

Then I calculated the person correlation coefficient between the selected user and all other user from the following equation:

$$\mathrm{Corr}(U_1, U_2) = \frac{\sum (r_{1,i} - \bar{r}_1)(r_{2,i} - \bar{r}_2)}{\sqrt{\sum (r_{1,i} - \bar{r}_1)^2}\sqrt{\sum (r_{2,i} - \bar{r}_2)^2}},$$

Corr(EN, LN) = $\frac{(4-3.75)(4-3)+(4-3.75)(4-3)+(2-3)(3-3.75)}{\sqrt{(4-3.75)^2+(4-3.75)^2+(3-3.75)^2}\sqrt{(4-3.75)^2(4-3)^2+(4-3)^2+(2-3)^2}} = 0.870$

Corr(EN,MH)= -1

Corr(EN,JH)=0

Corr(EN,DU)=0

Corr(EN,FL)=0

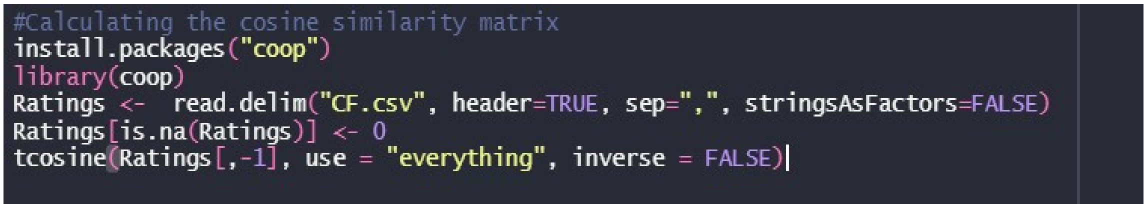
Corr(EN,GL)=0

Corr(EN,AH)=0

Corr(EN,SA)=0

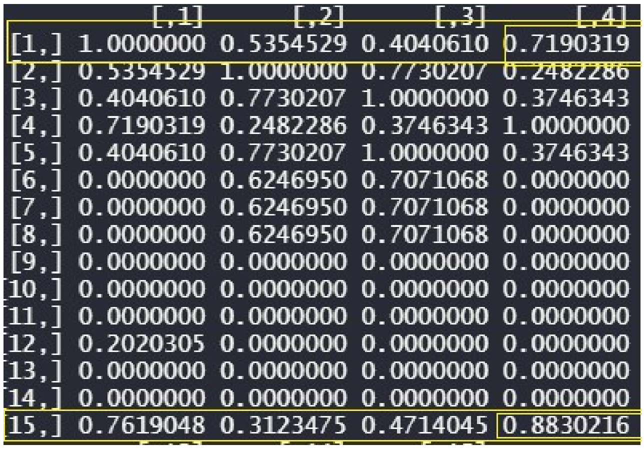
Corr(EN,RW)=0
Corr(EN,MG)=0
Corr(EN,AF)=0
Corr(EN,KG)=0
Corr(EN,DS)=0

2) Based on the previous calculations the single course that is to be recommended to the user EN is **Python** since it had the highest rating from the user LN who is the closest user to EN.
3) I used “coop” package to calculate the cosine similarity and I replaced the NA values with 0 (this is a little bit inaccurate as this suppose that all students give 0 rating to all courses they have not studied yet) and after this I constructed the cosine similarity matrix as the following



```
#Calculating the cosine similarity matrix
install.packages("coop")
library(coop)
Ratings <-  read.delim("CF.csv", header=TRUE, sep=",", stringsAsFactors=FALSE)
Ratings[is.na(Ratings)] <- 0
tcosine(Ratings[,-1], use = "everything", inverse = FALSE)
```

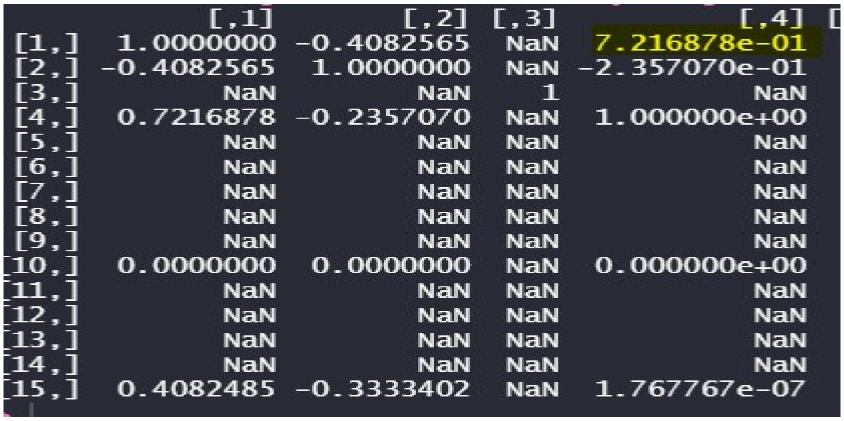
With regard to EN student (i.e. user number 4)



```
          [,1]      [,2]      [,3]      [,4]
 [1,] 1.0000000 0.5354529 0.4040610 0.7190319
 [2,] 0.5354529 1.0000000 0.7730207 0.2482286
 [3,] 0.4040610 0.7730207 1.0000000 0.3746343
 [4,] 0.7190319 0.2482286 0.3746343 1.0000000
 [5,] 0.4040610 0.7730207 1.0000000 0.3746343
 [6,] 0.0000000 0.6246950 0.7071068 0.0000000
 [7,] 0.0000000 0.6246950 0.7071068 0.0000000
 [8,] 0.0000000 0.6246950 0.7071068 0.0000000
 [9,] 0.0000000 0.0000000 0.0000000 0.0000000
[10,] 0.0000000 0.0000000 0.0000000 0.0000000
[11,] 0.0000000 0.0000000 0.0000000 0.0000000
[12,] 0.2020305 0.0000000 0.0000000 0.0000000
[13,] 0.0000000 0.0000000 0.0000000 0.0000000
[14,] 0.0000000 0.0000000 0.0000000 0.0000000
[15,] 0.7619048 0.3123475 0.4714045 0.8830216
```

4) From the previous result we can find that student 15 (DS) is the closest student to student 4, yet this student has not registered any subject other than the subjects that user four registered so, we moved to the next closest student which was user 1 LN Therefore, the recommendation would be **Python**.

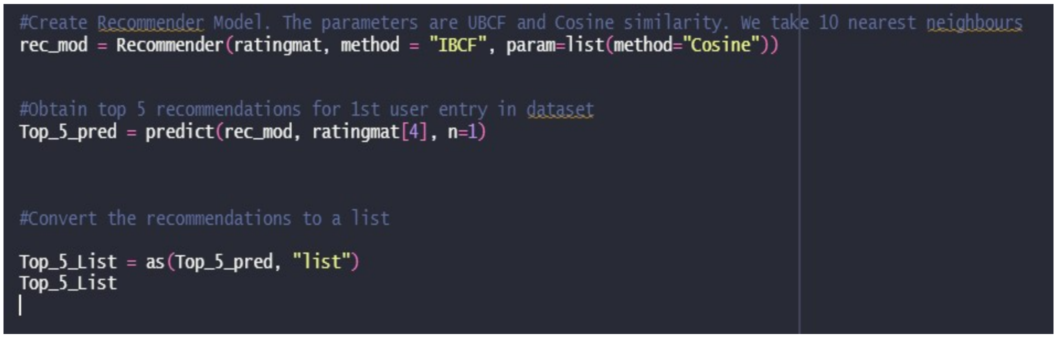
Another sol: I subtracted the average ratings for each user from all of the user ratings that way we can consider the 0 as the neutral value. After doing so, the resulted cosine similarity matrix was



```
               [,1]       [,2] [,3]          [,4] [
 [1,]  1.0000000 -0.4082565  NaN  7.216878e-01
 [2,] -0.4082565  1.0000000  NaN -2.357070e-01
 [3,]        NaN        NaN    1           NaN
 [4,]  0.7216878 -0.2357070  NaN  1.000000e+00
 [5,]        NaN        NaN  NaN           NaN
 [6,]        NaN        NaN  NaN           NaN
 [7,]        NaN        NaN  NaN           NaN
 [8,]        NaN        NaN  NaN           NaN
 [9,]        NaN        NaN  NaN           NaN
[10,]  0.0000000  0.0000000  NaN  0.000000e+00
[11,]        NaN        NaN  NaN           NaN
[12,]        NaN        NaN  NaN           NaN
[13,]        NaN        NaN  NaN           NaN
[14,]        NaN        NaN  NaN           NaN
[15,]  0.4082485 -0.3333402  NaN  1.767767e-07
```

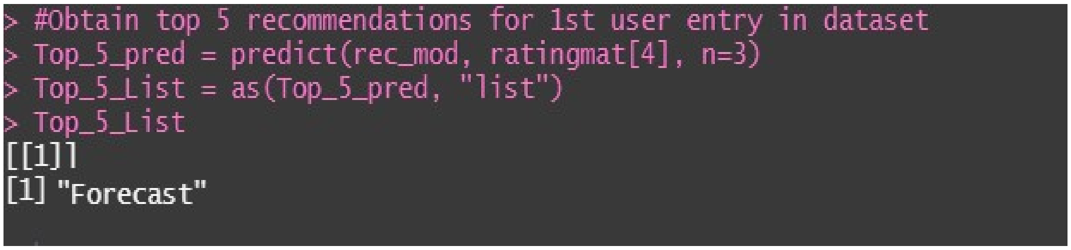
We can clearly conclude user LN is the closest user to EN therefore **Python** is the subject to be recommended to EN (same results obtain in the first solution)

5) I applied item-based rule using the following code



```
#Create Recommender Model. The parameters are UBCF and Cosine similarity. We take 10 nearest neighbours
rec_mod = Recommender(ratingmat, method = "IBCF", param=list(method="Cosine"))


#Obtain top 5 recommendations for 1st user entry in dataset
Top_5_pred = predict(rec_mod, ratingmat[4], n=1)



#Convert the recommendations to a list

Top_5_List = as(Top_5_pred, "list")
Top_5_List
|
```

Based on the results the **Forecast** subject is to be recommended to EN



```
> #Obtain top 5 recommendations for 1st user entry in dataset
> Top_5_pred = predict(rec_mod, ratingmat[4], n=3)
> Top_5_List = as(Top_5_pred, "list")
> Top_5_List
[[1]]
[1] "Forecast"
```