DTI 5126: Fundamentals for Applied Data Science

Summer 2021 - Assignmen 4

Name: Yomna Jehad Abdelsattar

Part A: Association Rules

I) a) Find all frequent itemsets in database X.

a)

Items	Frequency	Support	
A	5	5/8	=0.625
B	4	4/8	=0.5
C	5	5/8	=0.625
D	6	6/8	= 0.75
E	1	1/8	= 0.125 → <0.25 ∴Out
F	4	4/8	= 0.5
G	5	5/8	=0.625

Support = 25%

Itempairs	Frequency	Support	
A,B	3	0.375	
A,C	3	0.375	
A,D	4	0.5	
A,F	2	0.25	
A,G	2	0.25	
B,C	2	0.25	
B,D	2	0.25	
B,F	1	0.125	→ <0.25
B,G	2	0.25	
C,D	4	0.5	
C,F	2	0.25	
C,G	3	0.375	
D,F	4	0.5	
D,G	3	0.375	
F,G	2	0.25	

Item Pairs	Frequency	Support
A, B, C	1	0.125 → < 0.25
A, B, D	2	0.25
A, B, F	0	0 → < 0.25
A, B, G	1	0.125 → < 0.25
A, C, D	3	0.375
A, C, F	1	0.125 → < 0.25
A, C, G	1	0.125 → < 0.25
A, D, F	2	0.25
A, D, G	1	0.125 → < 0.25
A, F, G	0	0 → < 0.25
B, C, D	1	0.125 → < 0.25
B, C, G	1	0.125 → < 0.25
B, D, G	0	0 → < 0.25
C, D, F	2	0.25
C, D, G	2	0.25
C, F, G	1	0.125 → < 0.25
D, F, G	2	0.25

Item pairs	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

∵ C, D, FG has support = 0.125

∴ We stop at this iteration

& this table is our list of most frequent items

b) Find strong association rules for database X.

b) Association Rules

Conf. = 60%

1) A, B, D

A, B → D Confidence = 2/3 = 0.6667 > 0.6 ✓

A, D → B Conf. = 2/3 = 0.5 ×

D, B → A Conf. = 2/2 = 1 ✓

A → D, B Conf. = 2/5 = 0.4 ×

B → A, D Conf. = 2/4 = 0.5 ×

D → A, B Conf. = 2/6 = 1/3 ×

2) A, D, F

A, D → F Conf. = 2/4 = 0.5 ×

A, F → D Conf. = 2/2 = 1 ✓

D, F → A Conf. = 2/4 = 0.5 ×

A → D, f Conf. = 2/5 = 0.4 ×

D → A, f Conf. = 2/6 = 1/3 ×

F → A, D Conf. = 2/4 = 0.5 ×

3) C, D, F

C, D → F Conf. = 2/4 = 0.5

C, F → D Conf. = 2/2 = 1 ✓

F, D → C Conf. = 2/4 = 0.5

C → D, f Conf. = 2/5 = 0.4

D → C, F Conf. = 2/6 = 1/3

f → D, C Conf. = 2/4 = 0.5

4) C, D, G

C, D → G Conf. = 2/8 / 4/8 = 0.5

C, G → D Conf. = 2/8 / 3/8 = 0.667 ✓

D, G → C Conf. = 2/8 / 3/8 = 0.667 ✓

C → D, G Conf = 2/8 / 5/8 = 0.4

D → C, G Conf = 2/8 / 6/8 = 1/3

G → C, D Conf = 2/8 / 5/8 = 0.4

5) D, F, G

D, F → G Conf. = 2/8 / 4/8 = 0.5

F, G → D 2/8 / 2/8 = 1 ✓

D, G → F 2/8 / 3/8 = 0.667 ✓

D → F, G 2/8 / 6/8 = 1/3

F → D, G 2/8 / 4/8 = 0.5

G → D, F 2/8 / 5/8 = 0.4

6) A, C, D

A, C → D Conf. = 3/8 / 3/8 = 1 ✓

A, D → C = 3/8 / 4/8 = 0.75 ✓

C, D → A 3/8 / 4/8 = 0.75 ✓

A → C, D 3/8 / 5/8 = 0.6 ✓

C → A, D 3/8 / 5/8 = 0.6 ✓

D → A, C 3/8 / 6/8 = 0.5

The Strong Association Rules ARE :

A, B → D & D, B → A & D, G1 → C

A, F → D & C, D → A & A, D → C

C, F → D & F, G1 → D & A → C, D

C, G1 → D & D, G1 → F & C → A, D

A, C → D

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

c) $lift(A,B \rightarrow D) = \frac{2/8}{3/8 \times 6/8} = 0.89$

$lift(A,F \rightarrow D) = \frac{2/8}{2/8 \times 6/8} = 1.33$ ✓

$lift(C,F \rightarrow D) = \frac{2/8}{2/8 \times 6/8} = 1.33$ ✓

$lift(C,G \rightarrow D) = \frac{2/8}{3/8 \times 6/8} = 0.89$

$lift(A,C \rightarrow D) = \frac{3/8}{3/8 \times 6/8} = 1.33$ ✓

$lift(F,G \rightarrow D) = \frac{2/8}{2/8 \times 6/8} = 1.33$ ✓

$lift(D,B \rightarrow A) = \frac{2/8}{2/8 \times 5/8} = 1.6$ ✓

$lift(C,D \rightarrow A) = \frac{3/8}{4/8 \times 5/8} = 1.2$ ✓

$lift(D,G \rightarrow F) = \frac{2/8}{3/8 \times 4/8} = 1.33$ ✓

$lift(D,G \rightarrow C) = \frac{2/8}{3/8 \times 5/8} = 1.067$ ✓

$lift(A,D \rightarrow C) = \frac{3/8}{4/8 \times 5/8} = 1.2$ ✓

$lift(A \rightarrow C,D) = \frac{3/8}{5/8 \times 4/8} = 1.2$ ✓

$lift(C \rightarrow A,D) = \frac{3/8}{5/8 \times 4/8} = 1.2$ ✓

For lift Values $> 1 \Rightarrow$ item y is likely to be bought if item x is bought

∴ Misleading associations ARE:

$A,B \rightarrow D$ & $C,G \rightarrow D$

The Leading Associations ARE:

$A, F \rightarrow D$ + $C, F \rightarrow D$ + $A, C \rightarrow D$

$F, G \rightarrow D$ + $D, B \rightarrow A$ + $C, D \rightarrow A$

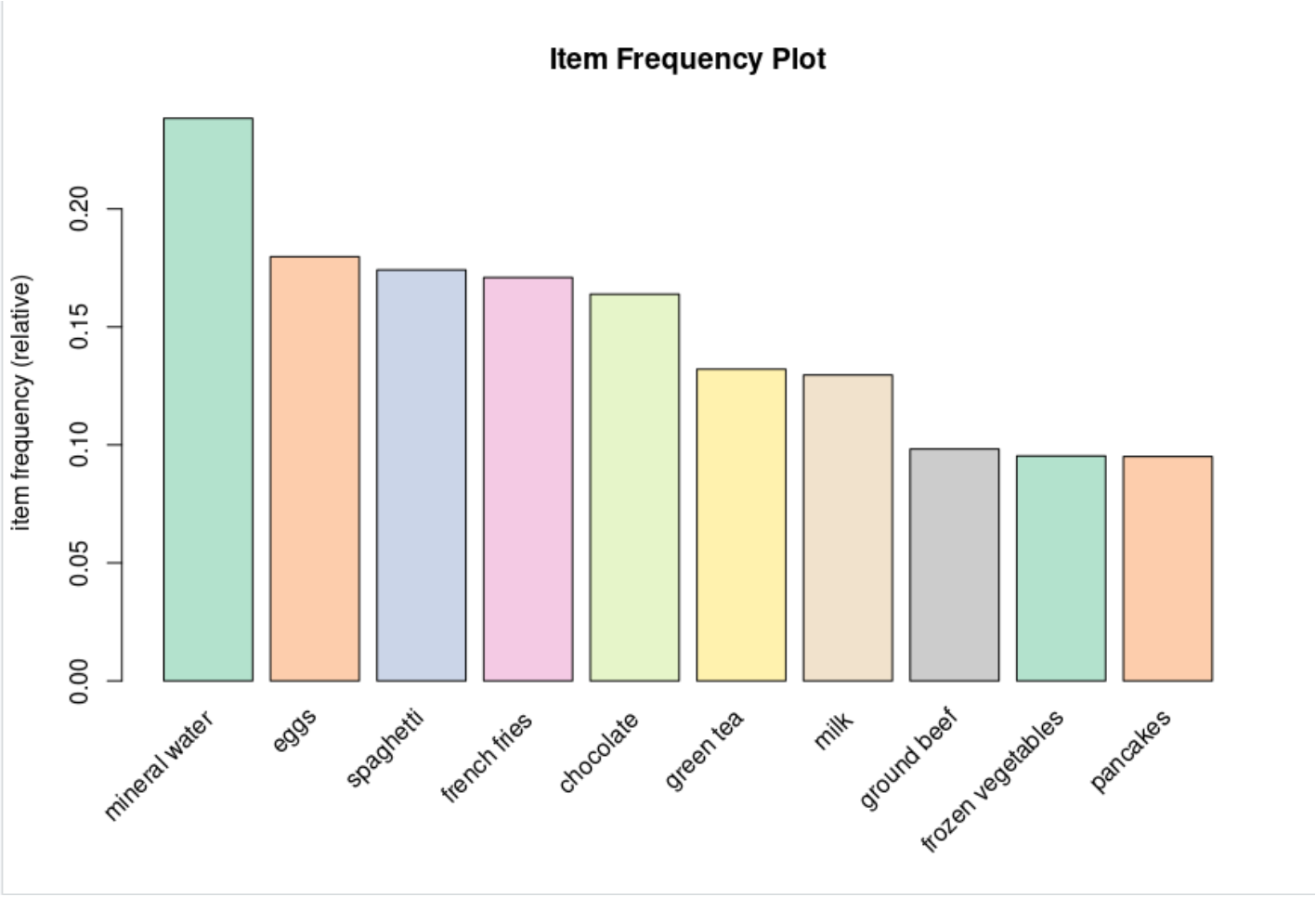
$D, G \rightarrow F$ + $D, G \rightarrow C$ + $A, D \rightarrow C$

$A \rightarrow C, D$ + $C \rightarrow A, D$

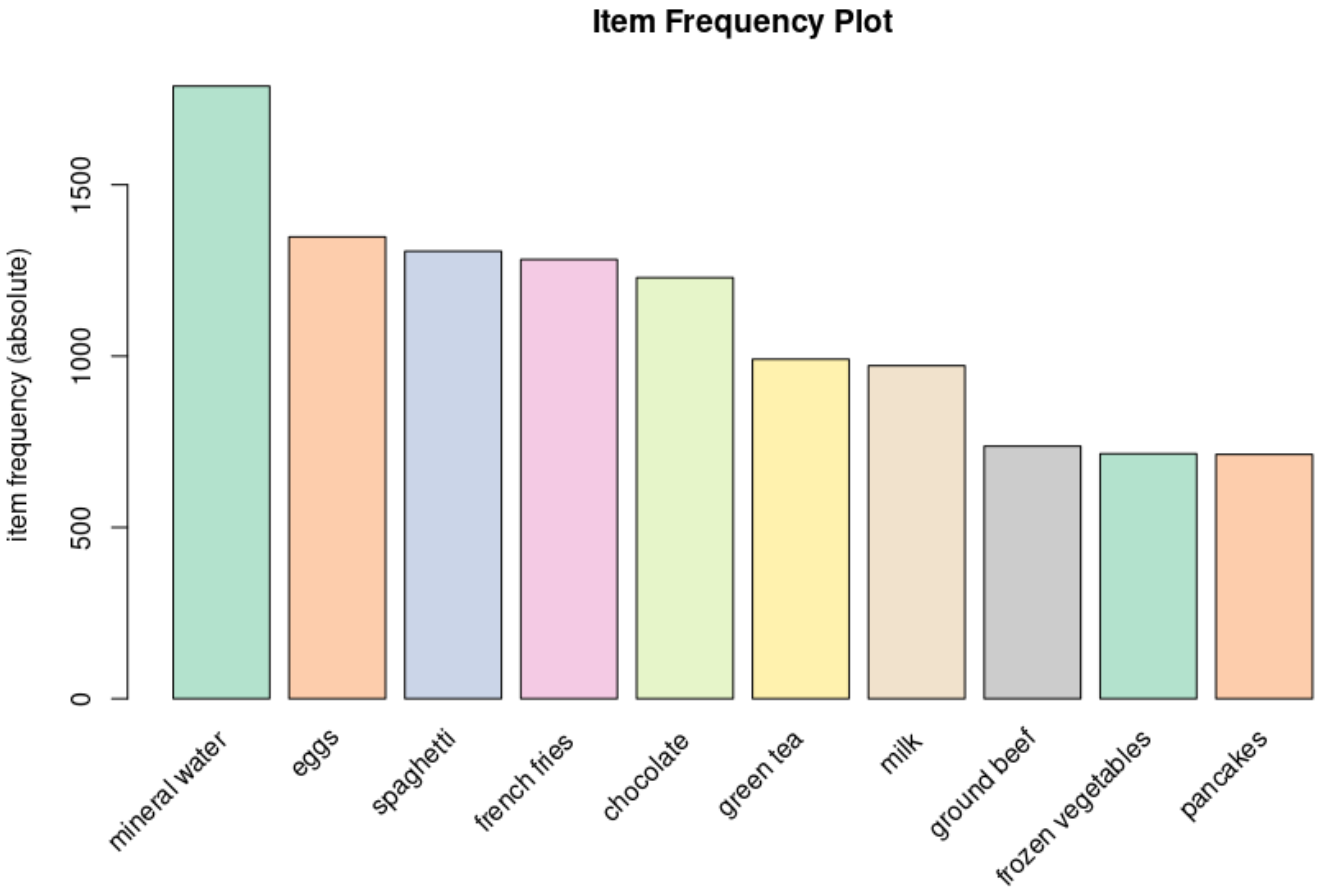
II)

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

- **Relative plot**



- **Absolute plot**



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.

- To generate the rules:

```
tr_rules <- apriori(tr, parameter = list(support =0.002, confidence = 0.2, maxlen = 3))
```

- To sort and display them:

```
> tr_rules_by_lift <- sort(tr_rules, by = "lift", decreasing=TRUE)
> inspect(head(tr_rules_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
>
```

- Top rule based on lift value:

```
> inspect(tr_rules_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
>
```

c) Select the rule from Q1 with the greatest lift. Compare this rule with the highest lift rule for maximum length of 2.

- Highest lift Rule with maxlen = 2

```
> tr_rules_2_by_lift_1 = tr_rules_2_by_lift[1]
> inspect(tr_rules_2_by_lift_1)
    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?

The Q1 rule (with a higher max length) has a better lift value. Which makes sense, the more rules explored, the better the chance to find a higher lift value.

ii) Which rule has the greater support?

The 2nd 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?

If I am a marketing manager, I would care about rules that occur **very frequently (high support)** and also that have a **very good relationship strength between items (high lift)** at the same time.
The first rule has a support of .0025 and a lift of 28
While the second rule has a support of.0033 (**slightly higher**) and a lift of 5.16 (**drastically lower**)
So since the supports have close values, we can say that both rules are similarly frequent. But the first rule has much more strength between its items. Which makes sense because escalope, mushrooms, cream sauce and pasta are items that form a well known meal. While formage blanc and honey do not specifically form a known meal.
In conclusion I would choose **the rule of Q1.**

Part B: Course Recommender System using Collaborative Filtering

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.?**

Q2) ① Students Correlated with E.N. : L.N., M.H, J.H., J.H. D.U., D.S.

⇒ Calculate Average of rating :

$EN : \frac{4+4+4+3}{4} = 3.75$

$LN: \frac{4+3+2+4+2}{5} = 3$

$M.H: \frac{3+4+4}{3} = 3.67$

$JH: \frac{2+2}{2} = 2$

$DU: \frac{4+4}{2} = 4$

$DS: \frac{4+4+2}{3} = 3.3$

⇒ Calculate Correlation between Students :

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

$$= \frac{1.25}{\sqrt{3} \times \frac{\sqrt{11}}{4}} = 0.87$$

$$Corr(EN, MH) = \frac{(4-3.75)(3-3.67)}{\sqrt{(4-3.75)^2}\ \sqrt{(3-3.67)^2}} = \frac{-0.1675}{0.1675} = -1$$

$$Corr(EN, JH) = \frac{(4-3.75)(2-2)}{\sqrt{(4-3.75)^2}\ \sqrt{(2-2)^2}} = \frac{0}{0} \rightarrow \text{undefined}$$

$$Corr(EN, DU) = \frac{(4-3.75)(4-4)}{\sqrt{(4-3.75)^2}\ \sqrt{(4-4)^2}} = \frac{0}{0} \rightarrow \text{undefined}$$

$$Corr(EN, DS) = \frac{(4-3.75)(4-3.3) + (4-3.75)(2-3.3) + (4-3.75)(4-3.3)}{\sqrt{(4-3.75)^2 + (4-3.75)^2 + (4+3.75)^2} \sqrt{(4-3.3)^2 + (4-3.3)^2 + (2-3.3)^2}}$$

$$= \frac{0.025}{0.433 \times 1.634} = 0.035$$

* We can see that the only valid values are:

0.87 , -1 , 0.035

(EN, LN) (EN, MH) (EN, DS)

↳ Highest positive Correlation. Which means they have the same direction of increase or decrease.

2) Based on the single nearest student to E.N. Which single course should we recommend to E.N.?

Q2) [1] Phython.

- Due to the high positive Correlation between LN & EN
- We checked the courses which LN took but EN didn't.
  Python, Forecast
- But LN rated Python higher than Forecast
- Then EN will probably rate Python high too.

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

```
> #compute the cosine similarity between students.
> cosine_method<-cosine(t(ratingmat))
> cosine_method
          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]     [,10]     [,11]     [,12]
 [1,] 1.0000000 0.0000000 1.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000 0.0000000 0.7071068
 [2,] 0.0000000 1.0000000 0.0000000 0.0000000 0.7071068 0.0000000 1.0000000 1.0000000 0.7071068 0.0000000 0.0000000 0.0000000
 [3,] 1.0000000 0.0000000 1.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000 0.0000000 0.7071068
 [4,] 0.0000000 0.0000000 0.0000000 1.0000000 0.4714045 0.8830216 0.0000000 0.0000000 0.4714045 0.0000000 0.7619048 0.0000000
 [5,] 0.0000000 0.7071068 0.0000000 0.4714045 1.0000000 0.3746343 0.7071068 0.7071068 1.0000000 0.0000000 0.4040610 0.0000000
 [6,] 0.0000000 0.0000000 0.0000000 0.8830216 0.3746343 1.0000000 0.0000000 0.0000000 0.3746343 0.0000000 0.7190319 0.0000000
 [7,] 0.0000000 1.0000000 0.0000000 0.0000000 0.7071068 0.0000000 1.0000000 1.0000000 0.7071068 0.0000000 0.0000000 0.0000000
 [8,] 0.0000000 1.0000000 0.0000000 0.0000000 0.7071068 0.0000000 1.0000000 1.0000000 0.7071068 0.0000000 0.0000000 0.0000000
 [9,] 0.0000000 0.7071068 0.0000000 0.4714045 1.0000000 0.3746343 0.7071068 0.7071068 1.0000000 0.0000000 0.4040610 0.0000000
[10,] 1.0000000 0.0000000 1.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000 0.0000000 0.7071068
[11,] 0.0000000 0.0000000 0.0000000 0.7619048 0.4040610 0.7190319 0.0000000 0.0000000 0.4040610 0.0000000 1.0000000 0.2020305
[12,] 0.7071068 0.0000000 0.7071068 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.7071068 0.2020305 1.0000000
[13,] 0.0000000 0.6246950 0.0000000 0.3123475 0.7730207 0.2482286 0.6246950 0.6246950 0.7730207 0.0000000 0.5354529 0.0000000
[14,] 0.4472136 0.0000000 0.4472136 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.4472136 0.0000000 0.3162278
[15,] 1.0000000 0.0000000 1.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000 0.0000000 0.7071068
          [,13]     [,14]     [,15]
 [1,] 0.0000000 0.4472136 1.0000000
 [2,] 0.6246950 0.0000000 0.0000000
 [3,] 0.0000000 0.4472136 1.0000000
 [4,] 0.3123475 0.0000000 0.0000000
 [5,] 0.7730207 0.0000000 0.0000000
 [6,] 0.2482286 0.0000000 0.0000000
 [7,] 0.6246950 0.0000000 0.0000000
 [8,] 0.6246950 0.0000000 0.0000000
 [9,] 0.7730207 0.0000000 0.0000000
[10,] 0.0000000 0.4472136 1.0000000
[11,] 0.5354529 0.0000000 0.0000000
[12,] 0.0000000 0.3162278 0.7071068
[13,] 1.0000000 0.0000000 0.0000000
[14,] 0.0000000 1.0000000 0.4472136
[15,] 0.0000000 0.4472136 1.0000000
```

For reference the students ids that correspond to each numeric id:

	student_id
1	AF
2	AH
3	BA
4	DS
5	DU
6	EN
7	FL
8	GL
9	JH
10	KG
11	LN
12	MG
13	MH
14	RW
15	SA

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

- Create the model User based and specify nearest neighbors (say 4 because after investigating the cosine similarity values it seems that around 4 students are very similar to E.N.)

```
> #"UBCF" stands for User-Based Collaborative Filtering
> rec_mod = Recommender(ratingmat, method = "UBCF", param=list(method="Cosine",nn=3))
>
```

- Generate one recommendation for E.N.

```
> #Obtain top first recommendations for 6th student (E.N)
> Top_1_pred = predict(rec_mod, ratingmat[6], n=1)
>
> #Convert the recommendations to a list
> Top_1_List = as(Top_1_pred, "list")
> Top_1_List
[[1]]
[1] "5"
```

- It recommends course of IDid 5.
- To know the name of the course

```
> #Merge the courses ids with names to get titles and genres
> names=left_join(Top_1_df, courses, by="course_id")
>
> #Print the titles and genres
> names
  course_id course_name
1         5      Python
```

In short, the recommended course : **Python**

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

- Same as previous but set the model to **Item based (IBCF)**

```
> rec_mod_I = Recommender(ratingmat, method = "IBCF", param=list(method="Cosine"))
> rec_mod_I
Recommender of type ‘IBCF’ for ‘realRatingMatrix’
```

- The recommended course is: **Forecast**.

```
> names_I
  course_id course_name
1         6    Forecast
```