استاد: دکتر شاکری — محمدطاها فخاریان، ۸۱۰۱۹۸۴۴۹

تمرین‌های تشریحی:

سوال ۱:

الف) برای بدست آوردن تمام itemsetهای مکرر توسط apriori، لازم است که مجموعه‌های C و L را تشکیل دهیم:

$MinSupport = 60\% \rightarrow minsupport = 3$

$C_1 = \{A: 4, B: 3, C: 2, D: 3, E: 2, K: 1\}$

$L_1 = \{A: 4, B: 3, D: 3\}$

$C_2 = \{AB: 4, AD: 3, BD: 3\}$

$L_2 = \{AB: 4, AD: 3, BD: 3\}$

$C_3 = \{ABD: 3\}$

$L_3 = \{ABD: 3\}$

پس itemsetهای مکرر برابرند با:

$\{A: 4, B: 3, D: 3, AB: 4, AD: 3, BD: 3, ABD: 3\}$

ب) برای بدست آوردن Association Ruleهای قوی که با metarule داده شده مطابقت دارند، باید از itemset مکرر ABD استفاده کنیم. همه ruleها را امتحان می‌کنیم:

$\forall x \in transaction, buys(x, A)\ and\ buys(x, B) \rightarrow buys(x, D)$

$Confidence: \frac{3}{4} = 75\%, support = \frac{3}{4} = 75\%$

این rule قوی نیست.

$\forall x \in transaction, buys(x, A)\ and\ buys(x, D) \rightarrow buys(x, B)$

$Confidence: \frac{4}{4} = 100\%, support = \frac{3}{4} = 75\%$

این rule قوی است.

$$\forall x \in transaction,\ buys(x,\ B)\ and\ buys(x,\ D) \rightarrow buys(x,\ A)$$
$$Confidence:\ \frac{4}{4} = 100\%,\ support\ =\ \frac{3}{4} = 75\%$$

این rule قوی است.

سوال ۲:

الف) برای بررسی هم‌بستگی و استقلال خرید hot dogs و خرید hamburgers، از معیار lift استفاده می‌کنیم. برای محاسبه lift داریم:

$$lift = \frac{P(hotdogs \cup hamburgers)}{P(hotdogs)P(hamburgers)} = \frac{\frac{2000}{5000}}{\frac{3000}{5000} * \frac{2500}{5000}} = \frac{4}{3} = 1.33$$

می‌دانیم در صورتی که مقدار lift برای دو itemset برابر با یک باشد، دو itemset از هم مستقل خواهند بود. در صورتی که بزرگتر از یک باشد، هم‌بستگی مثبت دارند و در صورتی که کوچکتر از یک باشد، هم‌بستگی منفی دارند. در اینجا چون مقدار lift بزرگتر از یک است، رابطه هم‌بستگی خرید hot dogs و hamburgers یک رابطه هم‌بستگی مثبت است.

ب) برای محاسبه all-confidence و cosine داریم:

$$Allconf(hotdogs\ , hamburgers)\ = \frac{sup(hotdogs \cup hamburgers)}{max(sup(hotdogs), sup(hamburgers))} = \frac{2000}{3000} = 0.67$$
$$Cosine(hotdogs\ , hamburgers)\ = \frac{sup(hotdogs \cup hamburgers)}{\sqrt{sup(hotdogs) * sup(hamburgers)}} = \frac{2000}{\sqrt{3000*2500}} = 0.73$$

سوال ۳:

برای پیدا کردن itemsetهای مکرر با استفاده از constraint داده شده با استفاده از الگوریتم FPGrowth، ابتدا مقدار support را حساب می‌کنیم و itemsetهای غیرمکرر را حذف می‌کنیم و باقی itemsetها را برحسب value به صورت نزولی مرتب می‌کنیم. Water و Tea حذف می‌شوند:

Milk(3000)	3
Butter(2500)	2
Peanut(2300)	5
Chips(2000)	4
Cake(1500)	2
Cheese(1200)	3

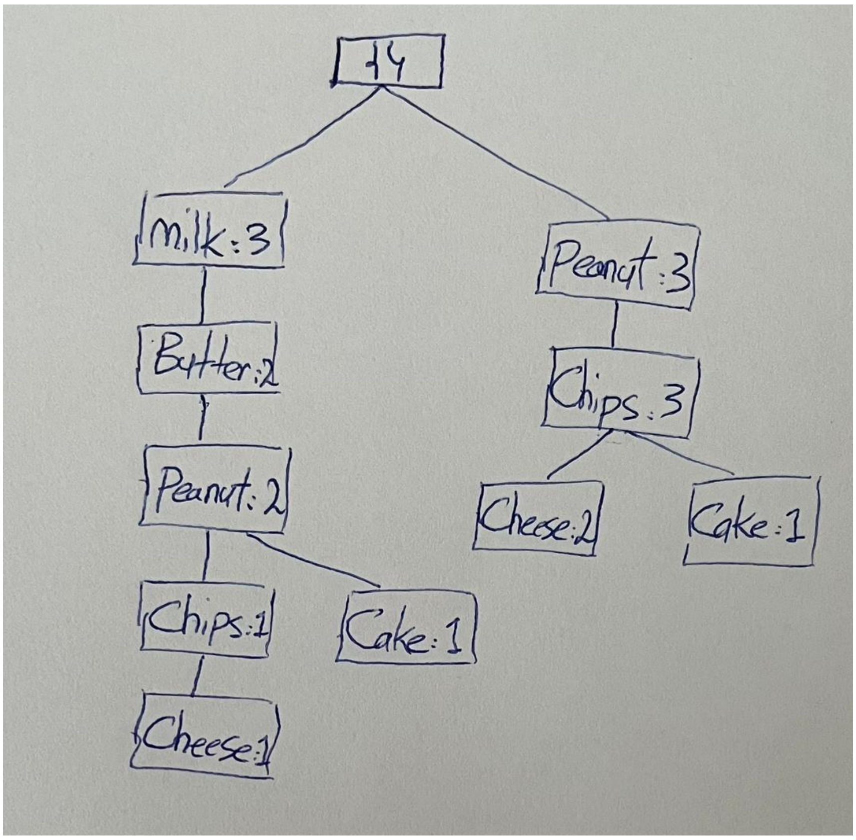
و جدول جدید به این صورت درمی‌آید:

Tid	Items	Freq items, ordered by value, desc
100	Milk, Peanut, Butter, Cake	Milk, Butter, Peanut, Cake
200	Cake, Chips, Peanut, Tea	Peanut, Chips, Cake
300	Cheese, Chips, Peanut	Peanut, Chips, Cheese
400	Chips, Milk, Cheese, Butter, Peanut	Milk, Butter, Peanut, Chips, Cheese
500	Milk, Water	Milk
600	Chips, Peanut, Cheese	Peanut, Chips, Cheese

و FP-list به این صورت خواهد بود:

FP-list: Milk - Butter - Peanut - Chips - Cake - Cheese

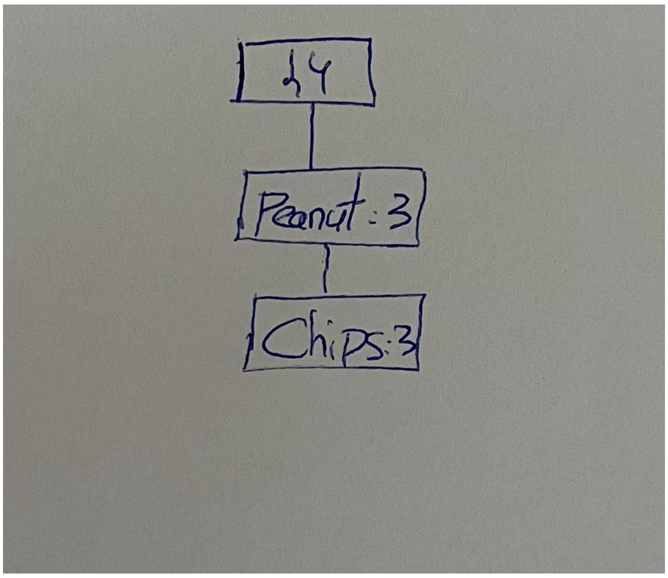
لذا FP-Tree به این صورت خواهد بود:



ابتدا Cheese-conditional pattern base را حساب می‌کنیم:

$$\{(Peanut,\ Chips): 2,\ (Milk,\ Butter,\ Peanut,\ Chips): 1\}$$

با حذف غیرمکررها به درخت جدید می‌رسیم:



چون درخت به یک مسیر تبدیل شده است، itemsetهای حاصل از cheese برابرند با:

$\{(Cheese)\colon 3,\ (Peanut,\ Cheese)\colon 3,\ (Chips,\ Cheese)\colon 3,\ (Peanut,\ Chips,\ Cheese)\colon 3\}$

حال Cake-conditional pattern base را حساب می‌کنیم:

$\{(Peanut,\ Chips)\colon 1,\ (Milk,\ Butter,\ Peanut)\colon 1\}$

با حذف غیرمکررها تنها یک $(Peanut)\colon 2$ می‌ماند. لذا itemsetهای حاصل از Cake برابرند با:

$\{(Cake)\colon 2,\ (Peanut,\ Cake)\colon 2\}$

نهایتاً هم Chips-conditional pattern base را حساب می‌کنیم:

$\{(Peanut,)\colon 3,\ (Milk,\ Butter,\ Peanut)\colon 1\}$

با حذف غیرمکررها تنها یک $(Peanut)\colon 4$ می‌ماند. لذا itemsetهای حاصل از Chips برابرند با:

$\{(Chips)\colon 4,\ (Peanut,\ Chips)\colon 4\}$

دقت کنید که لازم نیست باقی conditional pattern baseها ساخته شوند، زیرا دیگر در شرط داده شده صدق نمی‌کنند. لذا تمامی itemsetهای مکرر با شرایط گفته شده یافت شدند.

تمرین تشریحی امتیازی:

سوال ۱:

الف) برای اینکه مشخص کنیم کدام گره‌ها بازدید می‌شوند، چک می‌کنیم کدام گره‌ها با تابع هش داده شده مطابقت دارند:

$L_1\colon\ 1\ <\ 4\ and\ both\ in\ transaction\ \rightarrow gets\ visited$

$L_2\colon\ doesn't\ get\ visited\ because\ no\ combination\ is\ possible$

$L_3\colon\ 1\ <\ 5\ <\ 8\ and\ all\ in\ transation\ \rightarrow gets\ visited$

$L_4\colon\ doesn't\ get\ visited\ because\ no\ combination\ is\ possible$

$L_5\colon\ 1\ <\ 3\ \ and\ all\ in\ transation\ \rightarrow gets\ visited$

$L_6\colon\ doesn't\ get\ visited\ because\ no\ combination\ is\ possible$

$L_7\colon\ doesn't\ get\ visited\ because\ no\ combination\ is\ possible$

$L_8\colon\ doesn't\ get\ visited\ because\ no\ combination\ is\ possible$

L_9: $3 < 4$ *and all in transation* $\rightarrow$ *gets visited*

L_{11}: $3 < 5$ *and all in transation* $\rightarrow$ *gets visited*

L_{12}: *doesn't get visited because no combination is possible*

ب) با استفاده از گره‌های فوق، candidate itemsetها برابرند با:

$\{145, 158, 458\}$

تمرین‌های عملی:

In this computer assignment, we first analyze data for a market basket dataset and then, we try to mine frequent itemsets and association rules according to this data.

Question 1:

We first read the given csv file and check the first 5 rows to see how the given data look like:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice	frozen smoothie	spinach	olive oil
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

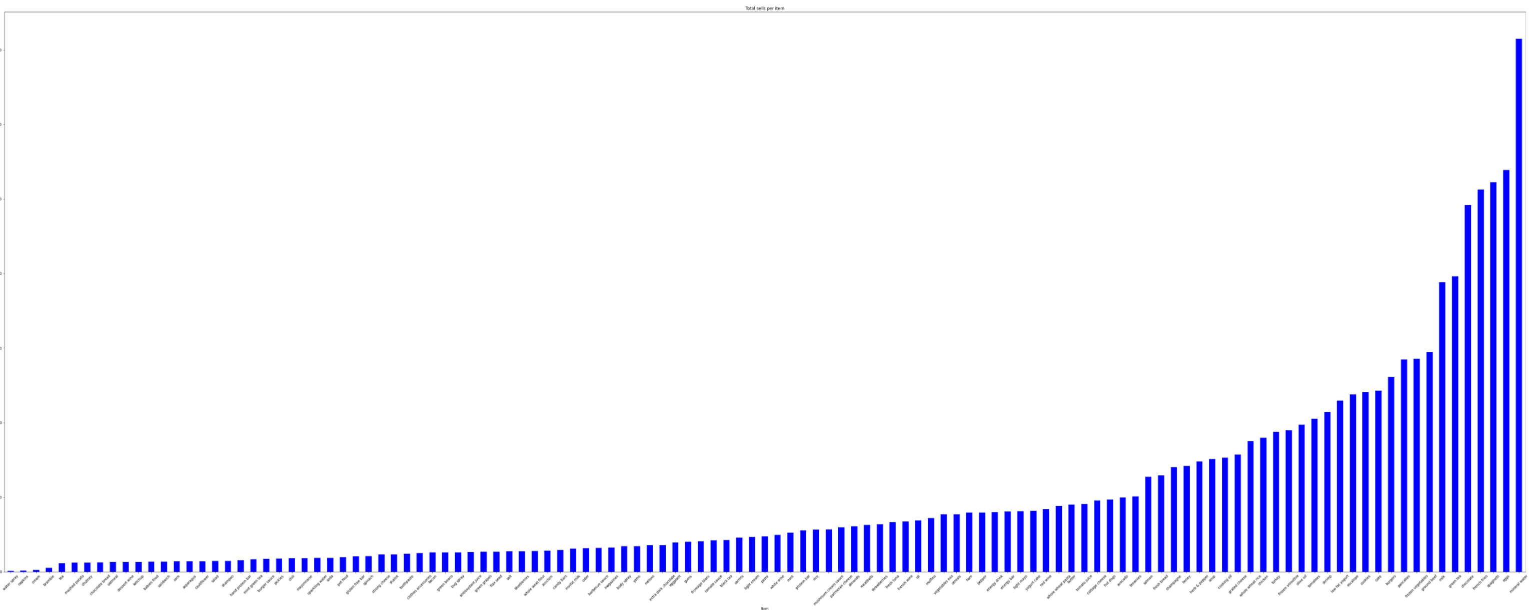
As you can see, we need to change the structure of the dataframe. First, as you can see, each row is a transaction in the market basket, but since some rows have more items, some rows have many NaN values, which is meaningless.

To do so, we need to make a dataframe that each column represents a unique item and each row represents a transaction and if a transaction contains an item, the value for that item and that transaction becomes True. Otherwise, it becomes False. We use 'TransactionEncoder' from mlxtend library to generate the new dataframe. There are more steps for preprocessing. If we check the items' names, we see that some items contain whitespaces at the beginning and ending of the name, which need to get removed. Also all items should get lowercase, so that same items but with different cases become similar in the end. Here is the result of preprocessing step according to these explanations:

	almonds	antioxydant juice	asparagus	avocado	babies food	bacon	barbecue sauce	black tea	blueberries	body spray	...	turkey	vegetables mix	water spray	white wine	whole weat flour	whole wheat pasta	whole wheat rice	yams	yogu ca
0	True	True	False	True	False	False	False	False	False	False	...	False	True	False	False	True	False	False	True	Fal
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	Fal
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	Fal
3	False	False	False	True	False	False	False	False	False	False	...	True	False	False	False	False	False	False	False	Fal
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	True	False	Fal
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7496	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	Fal
7497	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	Fal
7498	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	Fal
7499	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	Fal
7500	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	Tr

Pay attention! All columns and rows couldn't be shown in this picture.

Now let's plot the sells for each item and see which items are the most selling items:



As you can see, the most selling items in the market are the items that are used the most in everyday life like: mineral water, eggs and tomato. Also these items seem to be popular among a wide variety of people or are affordable for them, while items that are not sold that much seem to be some specific-purpose items like: babies food and hand protein bar. These items are not popular that much.

Question 2:

A:

The number of transactions are the number of rows in the new dataframe, which is 7501.

B:

The number of unique items are the number of columns in the new dataframe, which is 119.

C:

```
Top 5 most selling items with number of sells:
chocolate        1229
french fries     1282
spaghetti        1306
eggs             1348
mineral water    1788
```

D:

```
Number of transactions in which black tea is bought: 107
```

Question 3:

A:

Let's see the result using Apriori with different parameters:

- min-support = 0.003 and min-length = 2:

	support	itemsets	length
115	0.005199	(almonds, burgers)	2
116	0.003066	(almonds, cake)	2
117	0.005999	(almonds, chocolate)	2
118	0.006532	(eggs, almonds)	2
119	0.004399	(french fries, almonds)	2
...	...	...	...
1438	0.003066	(ground beef, spaghetti, pancakes, mineral water)	4
1439	0.003066	(tomatoes, spaghetti, ground beef, mineral water)	4
1440	0.003333	(milk, spaghetti, olive oil, mineral water)	4
1441	0.003066	(milk, spaghetti, shrimp, mineral water)	4
1442	0.003333	(tomatoes, milk, spaghetti, mineral water)	4

1328 rows × 3 columns

```
Min support = 0.003, Min length = 2 using Apriori. Spent time: 1.3110504150390625s. Total found = 1328
```

- min-support = 0.03 and min-length = 2:

	support	itemsets	length
36	0.033196	(eggs, chocolate)	2
37	0.034395	(french fries, chocolate)	2
38	0.032129	(milk, chocolate)	2
39	0.052660	(chocolate, mineral water)	2
40	0.039195	(spaghetti, chocolate)	2
41	0.036395	(eggs, french fries)	2
42	0.030796	(milk, eggs)	2
43	0.050927	(eggs, mineral water)	2
44	0.036528	(eggs, spaghetti)	2
45	0.033729	(french fries, mineral water)	2
46	0.035729	(frozen vegetables, mineral water)	2
47	0.031063	(green tea, mineral water)	2
48	0.040928	(ground beef, mineral water)	2
49	0.039195	(ground beef, spaghetti)	2
50	0.047994	(milk, mineral water)	2
51	0.035462	(milk, spaghetti)	2
52	0.033729	(pancakes, mineral water)	2
53	0.059725	(spaghetti, mineral water)	2

```
Min support = 0.03, Min length = 2 using Apriori. Spent time: 0.03237652778625488s. Total found = 18
```

- min-support = 0.3 and min-length = 2:

support	itemsets	length

```
Min support = 0.3, Min length = 2 using Apriori. Spent time: 0.005596160888671875s. Total found = 0
```

As you can see, as min-support increases, total number of frequent itemsets found decreases, in such way that for min-support = 0.3, algorithm couldn't find any frequent itemset. In contrast, if we decrease the min-support, some itemsets get mined that may not be interesting for us but satisfy the constraint. As you can see, when we put min-support = 0.003, 1328 itemsets get mined, which many of them are not that interesting for us. By using min-support = 0.03, we get some interesting itemsets that aren't small(18 frequent itemsets are enough).

B:

The best value for min-support is 0.03 according to the above explanation. Using 0.003 will result in mining itemsets that are not interesting for us and using 0.3 will result in mining no itemsets, but using 0.03 will result in enough and interesting itemsets, which is good for us.

C:

Let's see the result using FPGrowth with given parameters(28 itemsets are mined):

support	itemsets	length
0.238368	(mineral water)	1
0.132116	(green tea)	1

0.076523	(low fat yogurt)	1
0.071457	(shrimp)	1
0.065858	(olive oil)	1
0.063325	(frozen smoothie)	1
0.179709	(eggs)	1
0.087188	(burgers)	1
0.062525	(turkey)	1
0.129583	(milk)	1
0.058526	(whole wheat rice)	1
0.170911	(french fries)	1
0.050527	(soup)	1
0.174110	(spaghetti)	1
0.095321	(frozen vegetables)	1
0.080389	(cookies)	1
0.051060	(cooking oil)	1
0.163845	(chocolate)	1
0.059992	(chicken)	1
0.068391	(tomatoes)	1
0.095054	(pancakes)	1
0.052393	(grated cheese)	1
0.098254	(ground beef)	1

0.079323	(escalope)	1
0.081056	(cake)	1
0.050927	(eggs, mineral water)	2
0.059725	(spaghetti, mineral water)	2
0.052660	(chocolate, mineral water)	2

Question 4:

A:

Let’s see the result using FPGrowth with given parameters, sorted by lift in descending order(27 rules are mined):

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
(spaghetti)	(ground beef)	0.174110	0.098254	0.039195	0.225115	2.291162	0.022088	1.163716
(ground beef)	(spaghetti)	0.098254	0.174110	0.039195	0.398915	2.291162	0.022088	1.373997
(ground beef)	(mineral water)	0.098254	0.238368	0.040928	0.416554	1.747522	0.017507	1.305401
(frozen vegetables)	(mineral water)	0.095321	0.238368	0.035729	0.374825	1.572463	0.013007	1.218270
(spaghetti)	(milk)	0.174110	0.129583	0.035462	0.203675	1.571779	0.012900	1.093043
(milk)	(spaghetti)	0.129583	0.174110	0.035462	0.273663	1.571779	0.012900	1.137061
(milk)	(mineral water)	0.129583	0.238368	0.047994	0.370370	1.553774	0.017105	1.209650

(mineral water)	(milk)	0.238368	0.129583	0.047994	0.201342	1.553774	0.017105	1.089850
(milk)	(chocolate)	0.129583	0.163845	0.032129	0.247942	1.513276	0.010898	1.111823
(pancakes)	(mineral water)	0.095054	0.238368	0.033729	0.354839	1.488616	0.011071	1.180529
(mineral water)	(spaghetti)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008
(spaghetti)	(mineral water)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314
(spaghetti)	(chocolate)	0.174110	0.163845	0.039195	0.225115	1.373952	0.010668	1.079070
(chocolate)	(spaghetti)	0.163845	0.174110	0.039195	0.239219	1.373952	0.010668	1.085581
(chocolate)	(mineral water)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
(mineral water)	(chocolate)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256
(milk)	(eggs)	0.129583	0.179709	0.030796	0.237654	1.322437	0.007509	1.076009
(french fries)	(chocolate)	0.170911	0.163845	0.034395	0.201248	1.228284	0.006393	1.046827
(chocolate)	(french fries)	0.163845	0.170911	0.034395	0.209927	1.228284	0.006393	1.049383
(eggs)	(mineral water)	0.179709	0.238368	0.050927	0.283383	1.188845	0.008090	1.062815
(mineral water)	(eggs)	0.238368	0.179709	0.050927	0.213647	1.188845	0.008090	1.043158
(french fries)	(eggs)	0.170911	0.179709	0.036395	0.212949	1.184961	0.005681	1.042232
(eggs)	(french fries)	0.179709	0.170911	0.036395	0.202522	1.184961	0.005681	1.039640

(eggs)	(spaghetti)	0.179709	0.174110	0.036528	0.203264	1.167446	0.005239	1.036592
(spaghetti)	(eggs)	0.174110	0.179709	0.036528	0.209801	1.167446	0.005239	1.038081
(chocolate)	(eggs)	0.163845	0.179709	0.033196	0.202604	1.127397	0.003751	1.028711
(green tea)	(mineral water)	0.132116	0.238368	0.031063	0.235116	0.986357	-0.000430	0.995748

Top 3 rules by lift:

$\forall x \in transaction,\ buys(x,\ spaghetti) \rightarrow buys(x,\ ground\ beef)$

$\forall x \in transaction,\ buys(x,\ ground\ beef) \rightarrow buys(x,\ spaghetti)$

$\forall x \in transaction,\ buys(x,\ ground\ beef) \rightarrow buys(x,\ mineral\ water)$

B:

Let's see the result using FPGrowth with given parameters, sorted by lift in descending order(5 rules are mined):

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
4	(ground beef)	(spaghetti)	0.098254	0.174110	0.039195	0.398915	2.291162	0.022088	1.373997
3	(ground beef)	(mineral water)	0.098254	0.238368	0.040928	0.416554	1.747522	0.017507	1.305401
1	(frozen vegetables)	(mineral water)	0.095321	0.238368	0.035729	0.374825	1.572463	0.013007	1.218270
0	(milk)	(mineral water)	0.129583	0.238368	0.047994	0.370370	1.553774	0.017105	1.209650
2	(pancakes)	(mineral water)	0.095054	0.238368	0.033729	0.354839	1.488616	0.011071	1.180529

```
Min support = 0.03, Min confidence = 0.35 using FPGrowth. Spent time: 0.13147830963134766s. Total found = 5
```

Top 3 rules by lift:

$\forall x \in transaction,\ buys(x,\ ground\ beef) \rightarrow buys(x,\ spaghetti)$

$\forall x \in transaction,\ buys(x,\ ground\ beef) \rightarrow buys(x,\ mineral\ water)$

$\forall x \in transaction,\ buys(x,\ frozen\ vegetables) \rightarrow buys(x,\ mineral\ water)$

The number of mined rules decreases, since fewer rules can satisfy the new constraint(min-confidence has increased in this part) and because of that, fewer rules can get mined in this part.