

Прізвище: **КИРИЛЮК**
Ім'я: **Дмитро**
Група: **ПП-22**
Варіант: **08**
Дата захисту: **24.03.2025р.**



Кафедра: **САПР**
Дисципліна: **Системи інтелектуального аналізу та візуалізації даних**
Перевірив: **Андрій КЕРНИЦЬКИЙ**

ЗВІТ

до лабораторної роботи №06
на тему **“Асоціативний аналіз. Методи побудови асоціативних правил.”**

Мета роботи: ознайомлення з методами пошуку асоціативних правил за допомогою алгоритмів Apriori та FPGrowth в середовищах Weka та Excel. Студенти мають набути навичок роботи з цими алгоритмами, виконавши певні тренувальні завдання в середовищі Weka, а потім застосувати ці навички для виконання індивідуальних завдань на власних наборах даних. У програмі Excel студенти повинні застосувати алгоритм Apriori до свого власного набору даних про покупки для знаходження двійкових частих наборів

Вправа 1 - Датасет для голосування

Завантажте набір даних `vote.arff`.

- Завдання A1: запустіть Apriori, використовуючи налаштування параметрів за замовчуванням. Достовірність правила 10 становить 0.96. Як було обчислено це значення впевненості? Запишіть пропорцію у вигляді ділення.
- Завдання A2: Скільки випадків підтримує правило 8?
- Завдання A3: Що означає «правило застосовується до певної кількості випадків»? Поясніть на прикладі правила номер 7. (Підказка: ви можете перевірити числа на панелі попередньої обробки.)
- Завдання A4: Що означає «кількість випадків, передбачених правильно»? Поясніть на прикладі правила номер 9.
- Завдання A5: Вивчіть опис параметрів для Apriori, натиснувши кнопку 'More' у вікні, яке дозволяє вам змінити параметри для 'Apriori'. Спробуйте змінити кількість правил, указаних у вихідних даних. Як ви думаєте, чи може кількість згенерованих правил перевищувати 100. Якщо так, то чому?
- Завдання A6: Що означає «найкращі правила»? Який критерій використовується для визначення найкращих правил?
- Завдання A7: Яке правило визначає, наскільки ймовірно, що якщо конгресмен не голосував за допомогу Сальвадору, він також голосував за допомогу нікарагуанським контрабандистам?
- Завдання A8: 10 найкращих правил містять правила, які містять «Class=democrat» у правій частині. Чи говорить це щось про виборчі звички конгресменів-демократів?

vote.arff - Цей набір даних містить інформацію про те, як кожен із конгресменів Палати представників США голосував за 16 ключових законів. Один екземпляр представляє історію голосування одного конгресмена та його партій.

Вправа 2 - Датасет для погоди

Завантажте набір даних ``weather.nominal.arff``.

- Завдання B1: Розгляньте правило: `temperature=hot ==> humidity=normal`. Чим підтверджується це правило? Скільки випадків застосовується до цього правила і яке значення достовірності? (Щоб відповісти на це запитання, відкрийте вікно «Перегляд» на панелі попередньої обробки.)
- Завдання B2: Розгляньте правило: `temperature=hot humidity=high ==> windy=TRUE`. Чим підтверджується це правило? Скільки випадків застосовується до цього правила і яке значення достовірності? Далі запишіть номери екземплярів, які підтримують правило, і кількість екземплярів, які застосовуються до цього правила.
- Завдання B3: Чи може правило мати перевірки двох (чи більше) атрибутів праворуч, як у прикладі нижче: `outlook=sunny temperature=cool ==> humidity=normal play=yes`

`weather.nominal.arff` - Це дуже маленький набір даних лише з номінальними атрибутами.

Вправа 3. - Датасет для супермаркету

Завантажте набір даних ``supermarket.arff``. Використовуйте Apriori для створення правил і використовуйте їх, щоб сказати щось про купівельні звички клієнтів супермаркету. Згенеруйте близько 30 правил. Також може бути цікаво створити правила з одним конкретним атрибутом у правій частині. Їх можна згенерувати, встановивши для першого параметра значення ``true``, а для другого параметра — значення індексу атрибута (індекси атрибутів для цього параметра починаються з 0, а не з 1), яке ви хочете бачити в правій частині правил.

- Завдання C1: Вивчіть кілька згенерованих правил і опишіть одне спостереження, яке, на вашу думку, було зроблено щодо купівельних звичок клієнтів супермаркету. Також запишіть відповідні правила для цього спостереження.
- Завдання C2: Опишіть друге спостереження, яке, на вашу думку, було зроблено щодо купівельних звичок клієнтів супермаркету. Також запишіть відповідні правила для цього спостереження.
- Завдання C3: Чи пропонують спостереження, зроблені вами в завданнях C1 і C2, якісь напрямки дій для менеджера супермаркету? Якщо так, то якими вони можуть бути?

`supermarket.arff` - Цей набір даних описує купівельні звички покупців супермаркету. Більшість атрибутів позначають одну конкретну групу предметів. Значення є ``t``, якщо клієнт купив товар поза асортиментом товарів, а в іншому випадку відсутній. Є один екземпляр на клієнта. Набір даних не містить атрибут класу, оскільки це не потрібно для вивчення асоціативних правил.

Індивідуальне завдання:

1. Виконайте наступні завдання для власного набору даних у Weka:

- Запустіть алгоритм пошуку асоціативних правил Apriori.
- Яке значення для порогу підтримки було використано в побудованій моделі? Яке значення для порогу достовірності було використано?
- Запишіть 10 найкращих знайдених правил, вкажіть для них значення підтримки та достовірності.
- Що позначають числа ліворуч і праворуч від стрілки в знайдених асоціативних правилах?

2. Виконайте наступні завдання для власного набору даних у Weka :

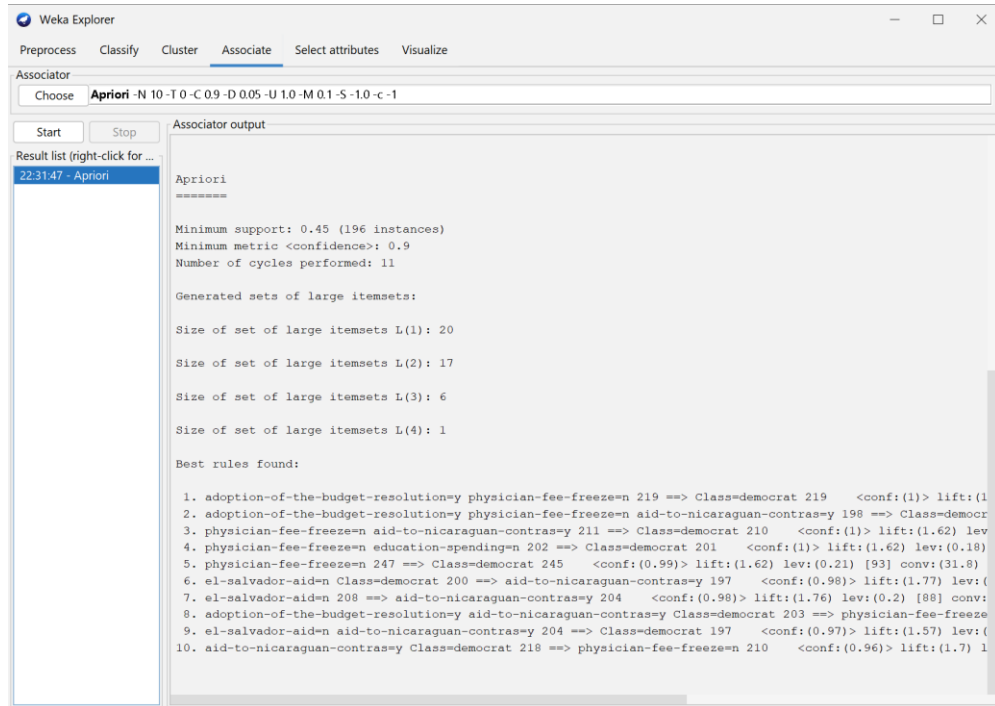
- Запустіть алгоритм пошуку асоціативних правил FPGrowth.

- Порівняйте списки десяти найкращих правил, отриманих двома алгоритмами. Поясніть відмінність в роботі двох алгоритмів.
- Згенеруйте також правила, у правій частині яких буде знаходитися ваш цільовий атрибут.

3. Виконайте наступні завдання для власного набору даних в Excel :

- Сформууйте власний набір транзакцій покупок, як показано у прикладі
- Знайдіть двоелементні групи частих наборів.
- Організуйте ваші обчислення на кількох аркушах (мінімум одніаркуш із вхідними даними і один аркуш для аналізу.)

Вправа 1



10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210 <conf:(0.96)> lift:(1.7) lev:(0.2) [86] conv:(10.47)

1) Впевненість (confidence) обчислюється за формулою:

$$Confidence = \frac{Support(X \cup Y)}{Support(X)}$$

Тобто це відношення випадків, коли зустрічаються X і Y разом, до випадків, коли зустрічається тільки X. У нашому випадку це 210/218 = 0.96.

2) Правило 8 підтримується у 203 випадках. Достовірність (conf): 0.98.

8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat 203 ==> physician-fee-freeze=n 198 <conf:(0.98)> lift:(1.72) lev:(0.19) [82] conv:(14.62)

3) Це означає, що в наборі даних є конкретна кількість випадків (рядків), які відповідають умові правила. Іншими словами, це кількість разів, коли ліва частина правила виконується в даних.

Приклад для правила №7:

- 208 конгресменів не голосували за допомогу Сальвадору (el-salvador-aid = n).
- З них 204 також голосували за допомогу Нікарагуа (aid-to-nicaraguan-contras = y).
- Впевненість правила (confidence) 0.98 означає, що 204 з 208 випадків підтверджують це правило.

Формула:

$$\text{Confidence} = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)} = \frac{204}{208} = 0.98$$

Отже, це правило застосовується до 208 випадків, тобто в 208 рядках датасету конгресмен не голосував за допомогу Сальвадору. І з них у 204 випадках він голосував за допомогу Нікарагуа, що робить правило дуже надійним (98%).

4) Правило говорить:

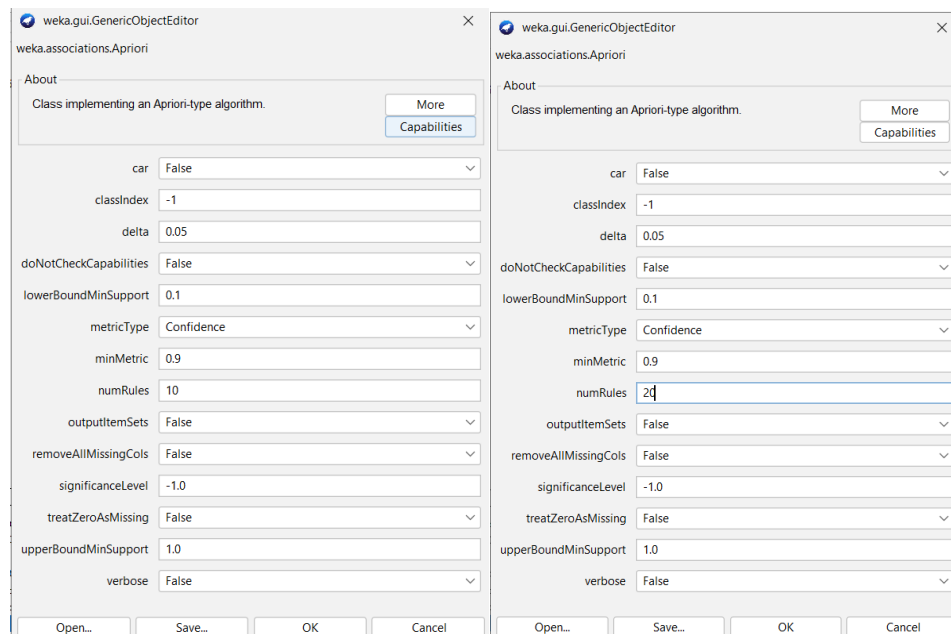
"Якщо проголосував за законопроект А, то проголосував за законопроект В".

Правило застосовується до 204 випадків у датасеті.

З них у 197 випадках передбачення правильне, тобто ці конгресмени дійсно є демократами.

Це означає, що правило має високу точність (97%) і є надійним для прогнозування партійної приналежності.

5)

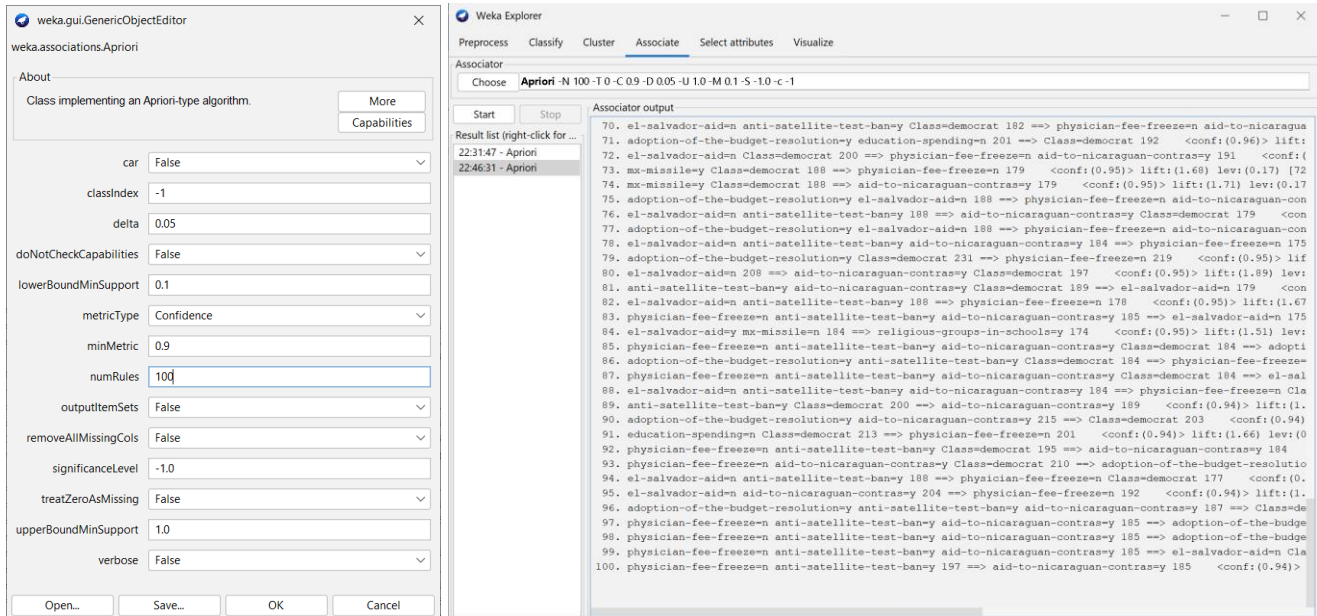


Чи може кількість згенерованих правил перевищувати 100? Чому?

Так, може, оскільки кількість правил залежить від:

- Мінімальної підтримки (min support) – якщо встановити дуже низьке значення, буде багато правил.
- Мінімальної довіри (min confidence) – якщо значення занадто низьке, генерується більше правил.
- Максимальної кількості правил (numRules) – цей параметр можна змінити.

Якщо параметр numRules збільшити, Weka може згенерувати більше ніж 100 правил.



6) Що означає «найкращі правила»?

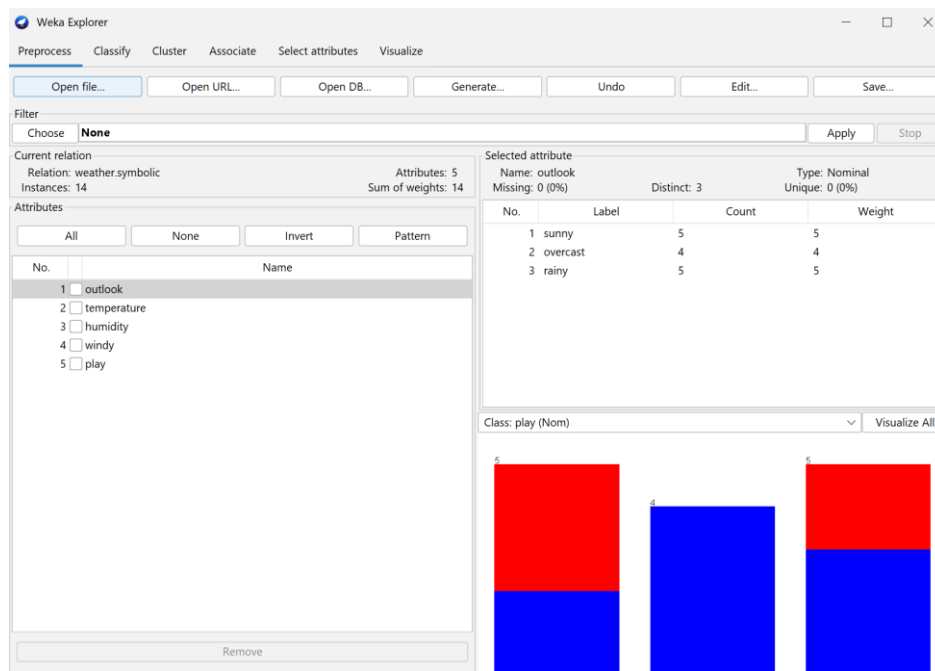
Weka вибирає найкращі правила за впевненістю (confidence) та підтримкою (support). Тобто:

- Чим вища підтримка, тим частіше зустрічається правило.
- Чим вища впевненість, тим точніше правило передбачає результат.

7) 7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204 <conf:(0.98)> lift:(1.76) lev:(0.2) [88] conv:(18.46)

8) У 10 найкращих правилах часто зустрічається Class=democrat у правій частині. Це свідчить про те, що багато голосувань демократів мають певні закономірності, і їхні голоси можна передбачити з високою точністю. Це означає, що демократи частіше голосували за певні законопроекти або ухвалювали рішення згідно з партійною лінією.

Вправа 2



Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Associator

Choose **Apriori** - N 10 - T 0 - C 0.9 - D 0.05 - U 1.0 - M 0.1 - S - 1.0 - c - 1

Start Stop

Result list (right-click for ...)

22:31:47 - Apriori

22:46:31 - Apriori

23:20:11 - Apriori

Associator output

=== Associator model (full training set) ===

Apriori

=====

Minimum support: 0.15 (2 instances)

Minimum metric <confidence>: 0.9

Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 47

Size of set of large itemsets L(3): 39

Size of set of large itemsets L(4): 6

Best rules found:

1. outlook=overcast 4 ==> play=yes 4 <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
2. temperature=cool 4 ==> humidity=normal 4 <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
3. humidity=normal windy=FALSE 4 ==> play=yes 4 <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
4. outlook=sunny play=no 3 ==> humidity=high 3 <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
5. outlook=sunny humidity=high 3 ==> play=no 3 <conf:(1)> lift:(2.8) lev:(0.14) [1] conv:(1.93)
6. outlook=rainy play=yes 3 ==> windy=FALSE 3 <conf:(1)> lift:(1.75) lev:(0.09) [1] conv:(1.29)
7. outlook=rainy windy=FALSE 3 ==> play=yes 3 <conf:(1)> lift:(1.56) lev:(0.08) [1] conv:(1.07)
8. temperature=cool play=yes 3 ==> humidity=normal 3 <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
9. outlook=sunny temperature=hot 2 ==> humidity=high 2 <conf:(1)> lift:(2) lev:(0.07) [1] conv:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2 <conf:(1)> lift:(2.8) lev:(0.09) [1] conv:(1.29)

Best rules found:

1. outlook=overcast 4 ==> play=yes 4 <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
2. temperature=cool 4 ==> humidity=normal 4 <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
3. humidity=normal windy=FALSE 4 ==> play=yes 4 <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
4. outlook=sunny play=no 3 ==> humidity=high 3 <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
5. outlook=sunny humidity=high 3 ==> play=no 3 <conf:(1)> lift:(2.8) lev:(0.14) [1] conv:(1.93)
6. outlook=rainy play=yes 3 ==> windy=FALSE 3 <conf:(1)> lift:(1.75) lev:(0.09) [1] conv:(1.29)
7. outlook=rainy windy=FALSE 3 ==> play=yes 3 <conf:(1)> lift:(1.56) lev:(0.08) [1] conv:(1.07)
8. temperature=cool play=yes 3 ==> humidity=normal 3 <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
9. outlook=sunny temperature=hot 2 ==> humidity=high 2 <conf:(1)> lift:(2) lev:(0.07) [1] conv:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2 <conf:(1)> lift:(2.8) lev:(0.09) [1] conv:(1.29)

Я розширив набір правил, зробивши аналіз глибшим, але ціною того, що частина правил має низьку підтримку (з'являються тільки в 1-2 випадках). Якщо мета – знайти стабільніші закономірності, варто підняти підтримку назад. Якщо мені треба знайти всі можливі залежності, то залишаю так, як є.

weka.gui.GenericObjectEditor

weka.associations.Apriori

About

Class implementing an Apriori-type algorithm.

More

Capabilities

car False

classIndex -1

delta 0.1

doNotCheckCapabilities False

lowerBoundMinSupport 0.05

metricType Confidence

minMetric 0.8

numRules 100

outputItemSets False

removeAllMissingCols False

significanceLevel -1.0

treatZeroAsMissing False

upperBoundMinSupport 0.9

verbose False

Open... Save... OK Cancel

1)
Перевіряємо список правил – такого немає.
Натомість є:

- temperature=hot humidity=normal 1 ==> outlook=overcast 1 (№63)
- temperature=hot humidity=normal 1 ==> windy=FALSE 1 (№73)

- temperature=hot humidity=normal 1 ==> play=yes 1 (№74)

Це означає, що temperature=hot не веде до humidity=normal у цьому наборі даних. Ймовірно, у вибірці humidity=high є більш типовим значенням для temperature=hot.

```
63. temperature=hot humidity=normal 1 ==> outlook=overcast 1 <conf:(1)> lift:(3.5) lev:(0.05) [0] conv:(0.71)
64. outlook=overcast temperature=mild 1 ==> humidity=high 1 <conf:(1)> lift:(2) lev:(0.04) [0] conv:(0.5)
65. outlook=overcast temperature=mild 1 ==> windy=TRUE 1 <conf:(1)> lift:(2.33) lev:(0.04) [0] conv:(0.57)
66. outlook=overcast temperature=mild 1 ==> play=yes 1 <conf:(1)> lift:(1.56) lev:(0.03) [0] conv:(0.36)
67. outlook=overcast temperature=cool 1 ==> humidity=normal 1 <conf:(1)> lift:(2) lev:(0.04) [0] conv:(0.5)
68. outlook=overcast temperature=cool 1 ==> windy=TRUE 1 <conf:(1)> lift:(2.33) lev:(0.04) [0] conv:(0.57)
69. outlook=overcast temperature=cool 1 ==> play=yes 1 <conf:(1)> lift:(1.56) lev:(0.03) [0] conv:(0.36)
70. temperature=cool play=no 1 ==> outlook=rainy 1 <conf:(1)> lift:(2.8) lev:(0.05) [0] conv:(0.64)
71. humidity=normal play=no 1 ==> outlook=rainy 1 <conf:(1)> lift:(2.8) lev:(0.05) [0] conv:(0.64)
72. temperature=hot windy=TRUE 1 ==> humidity=high 1 <conf:(1)> lift:(2) lev:(0.04) [0] conv:(0.5)
73. temperature=hot humidity=normal 1 ==> windy=FALSE 1 <conf:(1)> lift:(1.75) lev:(0.03) [0] conv:(0.43)
74. temperature=hot humidity=normal 1 ==> play=yes 1 <conf:(1)> lift:(1.56) lev:(0.03) [0] conv:(0.36)
```

2)

Правило: temperature=hot humidity=high ==> windy=TRUE

Шукаємо в списку – такого немає.

Натомість є схожі:

- temperature=hot humidity=high play=no 2 ==> outlook=sunny 2 (№35)
- temperature=hot humidity=high windy=TRUE 1 ==> outlook=sunny 1 (№80)
- outlook=sunny temperature=hot humidity=high 2 ==> play=no 2 (№37)

У випадку №80 правило справджується лише в одному екземплярі (support = 1). Це означає, що підтримка дуже низька.

Тобто, правило temperature=hot humidity=high ==> windy=TRUE не має значної підтримки в наборі.

```
35. temperature=hot humidity=high play=no 2 ==> outlook=sunny 2 <conf:(1)> lift:(2.8) lev:(0.09) [1] conv:(1.29)
36. outlook=sunny temperature=hot play=no 2 ==> humidity=high 2 <conf:(1)> lift:(2) lev:(0.07) [1] conv:(1)
37. outlook=sunny temperature=hot humidity=high 2 ==> play=no 2 <conf:(1)> lift:(2.8) lev:(0.09) [1] conv:(1.29)

80. temperature=hot humidity=high windy=TRUE 1 ==> outlook=sunny 1 <conf:(1)> lift:(2.8) lev:(0.05) [0] conv:(0.64)
```

3)

Чи можливе правило з декількома атрибутами праворуч?

Так, у цьому виводі з Weka є такі правила! Наприклад:

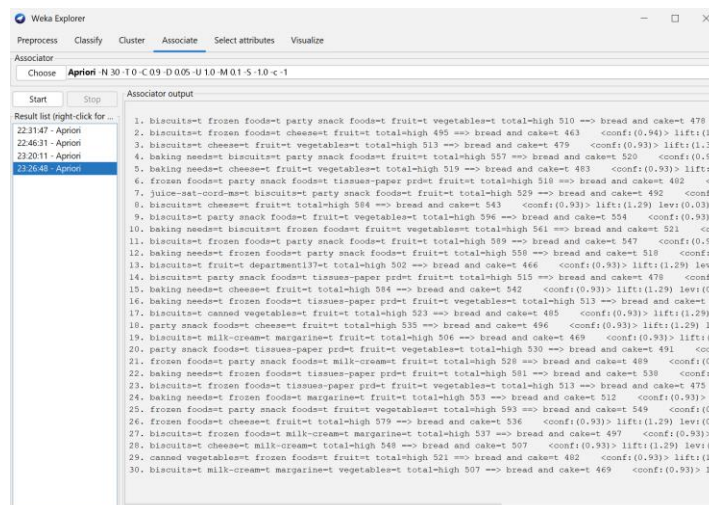
- temperature=hot play=no 2 ==> outlook=sunny humidity=high 2 (№38)
- outlook=sunny temperature=hot 2 ==> humidity=high play=no 2 (№39)

Отже, Weka може створювати правила з двома атрибутами праворуч, але такі випадки рідкісні.

Це залежить від підтримки в наборі даних.

```
38. temperature=hot play=no 2 ==> outlook=sunny humidity=high 2 <conf:(1)> lift:(4.67) lev:(0.11) [1] conv:(1.57)
39. outlook=sunny temperature=hot 2 ==> humidity=high play=no 2 <conf:(1)> lift:(3.5) lev:(0.1) [1] conv:(1.43)
```

Вправа 3



Best rules found:

```
1. biscuits=t fruit=t vegetables=t 1404 ==> bread and cake=t 1216 <conf:(0.87)> lift:(1.2) lev:(0.04) [205] conv:(2.08)
2. frozen foods=t fruit=t vegetables=t 1451 ==> bread and cake=t 1242 <conf:(0.86)> lift:(1.19) lev:(0.04) [197] conv:(1.94)
3. baking needs=t milk-cream=t fruit=t 1365 ==> bread and cake=t 1161 <conf:(0.85)> lift:(1.18) lev:(0.04) [178] conv:(1.87)
4. margarine=t fruit=t 1538 ==> bread and cake=t 1301 <conf:(0.85)> lift:(1.18) lev:(0.04) [194] conv:(1.81)
5. biscuits=t vegetables=t 1764 ==> bread and cake=t 1487 <conf:(0.84)> lift:(1.17) lev:(0.05) [217] conv:(1.78)
6. baking needs=t fruit=t vegetables=t 1489 ==> bread and cake=t 1255 <conf:(0.84)> lift:(1.17) lev:(0.04) [183] conv:(1.78)
7. tissues-paper prd=t milk-cream=t 1514 ==> bread and cake=t 1275 <conf:(0.84)> lift:(1.17) lev:(0.04) [185] conv:(1.77)
8. total=high 1679 ==> bread and cake=t 1413 <conf:(0.84)> lift:(1.17) lev:(0.04) [204] conv:(1.76)
9. biscuits=t milk-cream=t 1767 ==> bread and cake=t 1485 <conf:(0.84)> lift:(1.17) lev:(0.05) [213] conv:(1.75)
10. baking needs=t milk-cream=t vegetables=t 1392 ==> bread and cake=t 1169 <conf:(0.84)> lift:(1.17) lev:(0.04) [167] conv:(1.74)
11. biscuits=t fruit=t 1837 ==> bread and cake=t 1541 <conf:(0.84)> lift:(1.17) lev:(0.05) [218] conv:(1.73)
12. milk-cream=t margarine=t 1549 ==> bread and cake=t 1299 <conf:(0.84)> lift:(1.17) lev:(0.04) [184] conv:(1.73)
13. biscuits=t margarine=t 1493 ==> bread and cake=t 1251 <conf:(0.84)> lift:(1.16) lev:(0.04) [176] conv:(1.72)
14. milk-cream=t fruit=t vegetables=t 1571 ==> bread and cake=t 1311 <conf:(0.83)> lift:(1.16) lev:(0.04) [180] conv:(1.69)
15. biscuits=t frozen foods=t 1810 ==> bread and cake=t 1510 <conf:(0.83)> lift:(1.16) lev:(0.04) [207] conv:(1.69)
16. margarine=t vegetables=t 1587 ==> bread and cake=t 1322 <conf:(0.83)> lift:(1.16) lev:(0.04) [179] conv:(1.67)
17. biscuits=t tissues-paper prd=t 1453 ==> bread and cake=t 1209 <conf:(0.83)> lift:(1.16) lev:(0.04) [163] conv:(1.66)
18. frozen foods=t fruit=t 1861 ==> bread and cake=t 1548 <conf:(0.83)> lift:(1.16) lev:(0.05) [208] conv:(1.66)
19. frozen foods=t milk-cream=t 1826 ==> bread and cake=t 1516 <conf:(0.83)> lift:(1.15) lev:(0.04) [201] conv:(1.65)
20. party snack foods=t fruit=t 1592 ==> bread and cake=t 1321 <conf:(0.83)> lift:(1.15) lev:(0.04) [175] conv:(1.64)
21. tissues-paper prd=t vegetables=t 1559 ==> bread and cake=t 1293 <conf:(0.83)> lift:(1.15) lev:(0.04) [171] conv:(1.64)
22. baking needs=t milk-cream=t 1907 ==> bread and cake=t 1580 <conf:(0.83)> lift:(1.15) lev:(0.04) [207] conv:(1.63)
23. tissues-paper prd=t fruit=t 1567 ==> bread and cake=t 1297 <conf:(0.83)> lift:(1.15) lev:(0.04) [169] conv:(1.62)
24. party snack foods=t milk-cream=t 1541 ==> bread and cake=t 1275 <conf:(0.83)> lift:(1.15) lev:(0.04) [165] conv:(1.62)
25. milk-cream=t fruit=t 2038 ==> bread and cake=t 1684 <conf:(0.83)> lift:(1.15) lev:(0.05) [217] conv:(1.61)
26. baking needs=t margarine=t 1645 ==> bread and cake=t 1358 <conf:(0.83)> lift:(1.15) lev:(0.04) [174] conv:(1.6)
27. baking needs=t biscuits=t 1764 ==> bread and cake=t 1456 <conf:(0.83)> lift:(1.15) lev:(0.04) [186] conv:(1.6)
28. frozen foods=t tissues-paper prd=t 1505 ==> bread and cake=t 1239 <conf:(0.82)> lift:(1.14) lev:(0.03) [155] conv:(1.58)
29. baking needs=t fruit=t 1900 ==> bread and cake=t 1564 <conf:(0.82)> lift:(1.14) lev:(0.04) [196] conv:(1.58)
30. sauces-gravy-pkle=t fruit=t 1490 ==> bread and cake=t 1226 <conf:(0.82)> lift:(1.14) lev:(0.03) [153] conv:(1.58)
```

1) Спостереження:

Клієнти, які купують фрукти, овочі та печиво, часто також купують хліб і випічку.

Правила, що це підтверджують:

```
1. biscuits=t fruit=t vegetables=t 1404 ==> bread and cake=t 1216 <conf:(0.87)> lift:(1.2) lev:(0.04) [205] conv:(2.08)
5. biscuits=t vegetables=t 1764 ==> bread and cake=t 1487 <conf:(0.84)> lift:(1.17) lev:(0.05) [217] conv:(1.78)
6. baking needs=t fruit=t vegetables=t 1489 ==> bread and cake=t 1255 <conf:(0.84)> lift:(1.17) lev:(0.04) [183] conv:(1.78)
```

Висновок:

Це свідчить про те, що клієнти, які купують корисні продукти, також купують хлібобулочні вироби, можливо, для збалансованого харчування або приготування страв.

2) Спостереження:

Клієнти, які купують молочні продукти (молоко, вершки, маргарин), часто купують хліб і випічку.

Правила, що це підтверджують:

```
3. baking needs=t milk-cream=t fruit=t 1365 ==> bread and cake=t 1161 <conf:(0.85)> lift:(1.18) lev:(0.04) [178] conv:(1.87)
9. biscuits=t milk-cream=t 1767 ==> bread and cake=t 1485 <conf:(0.84)> lift:(1.17) lev:(0.05) [213] conv:(1.75)
12. milk-cream=t margarine=t 1549 ==> bread and cake=t 1299 <conf:(0.84)> lift:(1.17) lev:(0.04) [184] conv:(1.73)
```

Висновок:

Це може свідчити про те, що покупці купують молочні продукти разом із випічкою або для приготування домашніх десертів.

3) Оптимізація розташування товарів

- Розмістити хліб і випічку поруч із фруктами та овочами, а також біля печива, щоб підвищити ймовірність купівлі.
- Виділити окрему полицю для молочних продуктів (молоко, вершки, маргарин) поруч із хлібом і випічкою.

Спеціальні пропозиції та акції

- Зробити знижку "Купіть печиво + овочі + хліб зі знижкою 10%".

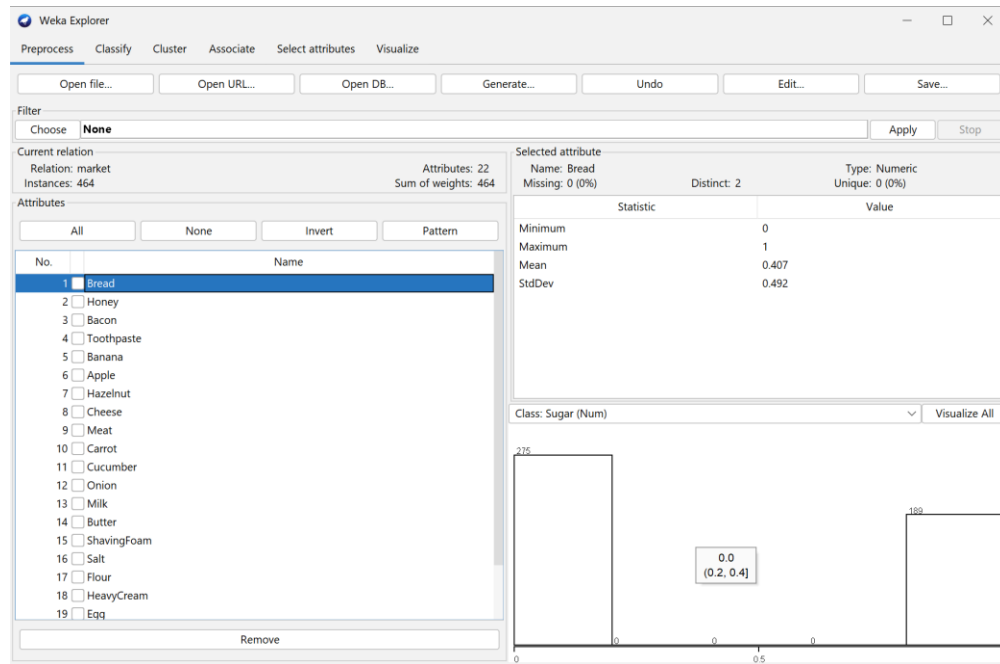
- Запропонувати комплекти для випічки (борошно, маргарин, молочні продукти + хліб) за зниженою ціною.

Персоналізовані рекомендації

- Для покупців, які часто купують молочні продукти, пропонувати знижки на хлібобулочні вироби через програму лояльності.

Індивідуальне завдання

Посилання на датасет: <https://www.kaggle.com/datasets/rukenmissonnier/real-market-data?resource=download&select=market.csv>



Weka Explorer - Associate tab

Associator: Choose **FPGrowth** - P 2 - I 1 - N 10 - T 0 - C 0.8 - D 0.05 - U 1.0 - M 0.1

Start Stop

Result list (right-click for ...)

- 02:30:19 - Apriori
- 02:30:32 - FPGrowth

Associator output

```

Sugar
=== Associator model (full training set) ===

Apriori
=====

Minimum support: 0.2 (93 instances)
Minimum metric <confidence>: 0.8
Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets L(1): 44
Size of set of large itemsets L(2): 638
Size of set of large itemsets L(3): 1540
Size of set of large itemsets L(4): 1509

Best rules found:

1. Hazelnut=n Carrot=n Onion=n 116 ==> Toothpaste=n 102 <conf:(0.88)> lift:(1.43) lev:(0.07) [30] conv:(2.97)
2. Hazelnut=n Onion=n Flour=n 116 ==> Toothpaste=n 102 <conf:(0.88)> lift:(1.43) lev:(0.07) [30] conv:(2.97)
3. Apple=n Hazelnut=n Onion=n 111 ==> Toothpaste=n 95 <conf:(0.86)> lift:(1.39) lev:(0.06) [26] conv:(2.5)
4. ShavingFoam=n Salt=n Flour=n 111 ==> Toothpaste=n 95 <conf:(0.86)> lift:(1.39) lev:(0.06) [26] conv:(2.5)
5. Bread=n Apple=n Cheese=n 113 ==> Butter=n 96 <conf:(0.85)> lift:(1.36) lev:(0.05) [25] conv:(2.35)
6. Honey=n Hazelnut=n Onion=n 113 ==> Toothpaste=n 96 <conf:(0.85)> lift:(1.38) lev:(0.06) [26] conv:(2.41)
7. Toothpaste=n Banana=n Cucumber=n 118 ==> Onion=n 100 <conf:(0.85)> lift:(1.37) lev:(0.06) [26] conv:(2.36)
8. Banana=n Butter=n Salt=n 117 ==> Toothpaste=n 99 <conf:(0.85)> lift:(1.37) lev:(0.06) [26] conv:(2.36)
9. Banana=n ShavingFoam=n Salt=n 110 ==> Toothpaste=n 93 <conf:(0.85)> lift:(1.37) lev:(0.05) [25] conv:(2.34)
10. Hazelnut=n Meat=n Carrot=n 110 ==> Toothpaste=n 93 <conf:(0.85)> lift:(1.37) lev:(0.05) [25] conv:(2.34)

```

1)

Поріг підтримки (Support):

Параметр lowerBoundMinSupport встановлений на значення 0.1, що означає, що мінімальна підтримка для правил дорівнює 10%.

Поріг достовірності (Confidence):

Параметр minMetric встановлений на 0.8, що означає, що мінімальний поріг достовірності (confidence) для правил дорівнює 80%.

Best rules found:

```
1. Hazelnut=n Carrot=n Onion=n 116 ==> Toothpaste=n 102    <conf:(0.88)> lift:(1.43) lev:(0.07) [30] conv:(2.97)
2. Hazelnut=n Onion=n Flour=n 116 ==> Toothpaste=n 102    <conf:(0.88)> lift:(1.43) lev:(0.07) [30] conv:(2.97)
3. Apple=n Hazelnut=n Onion=n 111 ==> Toothpaste=n 95     <conf:(0.86)> lift:(1.39) lev:(0.06) [26] conv:(2.5)
4. ShavingFoam=n Salt=n Flour=n 111 ==> Toothpaste=n 95     <conf:(0.86)> lift:(1.39) lev:(0.06) [26] conv:(2.5)
5. Bread=n Apple=n Cheese=n 113 ==> Butter=n 96          <conf:(0.85)> lift:(1.36) lev:(0.05) [25] conv:(2.35)
6. Honey=n Hazelnut=n Onion=n 113 ==> Toothpaste=n 96     <conf:(0.85)> lift:(1.38) lev:(0.06) [26] conv:(2.41)
7. Toothpaste=n Banana=n Cucumber=n 118 ==> Onion=n 100    <conf:(0.85)> lift:(1.37) lev:(0.06) [26] conv:(2.36)
8. Banana=n Butter=n Salt=n 117 ==> Toothpaste=n 99       <conf:(0.85)> lift:(1.37) lev:(0.06) [26] conv:(2.36)
9. Banana=n ShavingFoam=n Salt=n 110 ==> Toothpaste=n 93    <conf:(0.85)> lift:(1.37) lev:(0.05) [25] conv:(2.34)
10. Hazelnut=n Meat=n Carrot=n 110 ==> Toothpaste=n 93     <conf:(0.85)> lift:(1.37) lev:(0.05) [25] conv:(2.34)
```

Support (Підтримка): Кількість транзакцій, у яких виконується дане правило

Confidence (Достовірність): Ймовірність того, що права частина правила виконається, якщо виконані всі умови з лівої частини.

Lift: Показує, наскільки часто продукти в правилі трапляються разом порівняно з їх випадковим співпадінням. Значення Lift > 1 означає позитивну асоціацію (сильний зв'язок).

Conviction (Conv): Чим вище значення Conviction, тим сильніша впевненість, що правило не є випадковим.

- Ліва частина (до стрілки) – це умова або передумова (Antecedent). Вона вказує на набір продуктів чи атрибутів, які одночасно зустрічаються в певній кількості транзакцій.
- Права частина (після стрілки) – це наслідок або результат (Consequent). Вона показує, який товар або атрибут буде зустрічатися з високою ймовірністю за виконання умов лівої частини правила.

2)

FPGrowth found 19 rules (displaying top 10)

```
1. [Sugar=n, Apple=n, Bread=n, HeavyCream=n, Bacon=n]: 56 ==> [Egg=n]: 47    <conf:(0.84)> lift:(1.41) lev:(0.03) conv:(2.26)
2. [Sugar=n, Apple=n, Bread=n, Bacon=n, Banana=n]: 56 ==> [Egg=n]: 47    <conf:(0.84)> lift:(1.41) lev:(0.03) conv:(2.26)
3. [Sugar=n, Apple=n, Hazelnut=n, Bacon=n, Banana=n]: 59 ==> [Egg=n]: 49    <conf:(0.83)> lift:(1.39) lev:(0.03) conv:(2.16)
4. [Sugar=n, Apple=n, Bacon=n, Banana=n]: 81 ==> [Egg=n]: 67    <conf:(0.83)> lift:(1.39) lev:(0.04) conv:(2.18)
5. [Egg=n, Carrot=n, Hazelnut=n, Bacon=n]: 75 ==> [Banana=n]: 62    <conf:(0.83)> lift:(1.5) lev:(0.04) conv:(2.4)
6. [Sugar=n, Apple=n, Bread=n, Carrot=n, Bacon=n]: 57 ==> [Egg=n]: 47    <conf:(0.82)> lift:(1.38) lev:(0.03) conv:(2.09)
7. [Egg=n, Apple=n, Bread=n, Carrot=n, Bacon=n]: 57 ==> [Sugar=n]: 47    <conf:(0.82)> lift:(1.3) lev:(0.02) conv:(1.9)
8. [Egg=n, Apple=n, Bread=n, HeavyCream=n, Bacon=n]: 57 ==> [Sugar=n]: 47    <conf:(0.82)> lift:(1.3) lev:(0.02) conv:(1.9)
9. [Egg=n, Apple=n, HeavyCream=n, Hazelnut=n, Bacon=n]: 57 ==> [Sugar=n]: 47    <conf:(0.82)> lift:(1.3) lev:(0.02) conv:(1.9)
10. [Egg=n, Apple=n, HeavyCream=n, Bacon=n]: 79 ==> [Sugar=n]: 65    <conf:(0.82)> lift:(1.3) lev:(0.03) conv:(1.93)
```

Таблиця порівняння правил(за результатами):

№	Apriori (Правило → Результат, conf)	FP-Growth (Правило → Результат, conf)
1	Hazelnut=n, Carrot=n, Onion=n → Toothpaste=n (0.88)	Sugar=n, Apple=n, Bread=n, HeavyCream=n, Bacon=n → Egg=n (0.84)
2	Hazelnut=n, Onion=n, Flour=n → Toothpaste=n (0.88)	Sugar=n, Apple=n, Bread=n, Bacon=n, Banana=n → Egg=n (0.84)
3	Apple=n, Hazelnut=n, Onion=n → Toothpaste=n (0.86)	Sugar=n, Apple=n, Hazelnut=n, Bacon=n, Banana=n → Egg=n (0.83)
4	ShavingFoam=n, Salt=n, Flour=n → Toothpaste=n (0.86)	Sugar=n, Apple=n, Bacon=n, Banana=n → Egg=n (0.83)
5	Bread=n, Apple=n, Cheese=n → Butter=n (0.85)	Egg=n, Carrot=n, Hazelnut=n, Bacon=n → Banana=n (0.83)
6	Honey=n, Hazelnut=n, Onion=n → Toothpaste=n (0.85)	Sugar=n, Apple=n, Bread=n, Carrot=n, Bacon=n → Egg=n (0.82)
7	Toothpaste=n, Banana=n, Cucumber=n → Onion=n (0.85)	Egg=n, Apple=n, Bread=n, Carrot=n, Bacon=n → Sugar=n (0.82)
8	Banana=n, Butter=n, Salt=n → Toothpaste=n (0.85)	Egg=n, Apple=n, Bread=n, HeavyCream=n, Bacon=n → Sugar=n (0.82)
9	Banana=n, ShavingFoam=n, Salt=n → Toothpaste=n (0.85)	Egg=n, Apple=n, HeavyCream=n, Hazelnut=n, Bacon=n → Sugar=n (0.82)
10	Hazelnut=n, Meat=n, Carrot=n → Toothpaste=n (0.85)	Egg=n, Apple=n, HeavyCream=n, Bacon=n → Sugar=n (0.82)

Відмінності:

- Apriori частіше пов'язує Toothpaste з різними комбінаціями продуктів (Hazelnut, Onion, Flour тощо).
- FP-Growth частіше знаходить правила, що включають Egg та Sugar разом з іншими продуктами (Apple, Bread, Bacon, Banana).
- Apriori використовує окремі товари як основу для правил, а FP-Growth частіше включає набори з більшою кількістю товарів.

weka.gui.GenericObjectEditor

weka.associations.Apriori

About

Class implementing an Apriori-type algorithm.

More

Capabilities

carFalse

classIndex4

delta0.05

doNotCheckCapabilitiesFalse

lowerBoundMinSupport0.1

metricTypeConfidence

minMetric0.8

numRules10

outputItemSetsFalse

removeAllMissingColsFalse

significanceLevel-1.0

treatZeroAsMissingFalse

upperBoundMinSupport1.0

verboseFalse

Open...

Save...

OK

Cancel

Best rules found:

- Hazelnut=n Carrot=n Onion=n 116 ==> Toothpaste=n 102 <conf:(0.88)> lift:(1.43) lev:(0.07) [30] conv:(2.97)
- Hazelnut=n Onion=n Flour=n 116 ==> Toothpaste=n 102 <conf:(0.88)> lift:(1.43) lev:(0.07) [30] conv:(2.97)
- Apple=n Hazelnut=n Onion=n 111 ==> Toothpaste=n 95 <conf:(0.86)> lift:(1.39) lev:(0.06) [26] conv:(2.5)
- ShavingFoam=n Salt=n Flour=n 111 ==> Toothpaste=n 95 <conf:(0.86)> lift:(1.39) lev:(0.06) [26] conv:(2.5)
- Bread=n Apple=n Cheese=n 113 ==> Butter=n 96 <conf:(0.85)> lift:(1.36) lev:(0.05) [25] conv:(2.35)
- Honey=n Hazelnut=n Onion=n 113 ==> Toothpaste=n 96 <conf:(0.85)> lift:(1.38) lev:(0.06) [26] conv:(2.41)
- Toothpaste=n Banana=n Cucumber=n 118 ==> Onion=n 100 <conf:(0.85)> lift:(1.37) lev:(0.06) [26] conv:(2.36)
- Banana=n Butter=n Salt=n 117 ==> Toothpaste=n 99 <conf:(0.85)> lift:(1.37) lev:(0.06) [26] conv:(2.36)
- Banana=n ShavingFoam=n Salt=n 110 ==> Toothpaste=n 93 <conf:(0.85)> lift:(1.37) lev:(0.05) [25] conv:(2.34)
- Hazelnut=n Meat=n Carrot=n 110 ==> Toothpaste=n 93 <conf:(0.85)> lift:(1.37) lev:(0.05) [25] conv:(2.34)

Генерація таких правил дозволяє аналізувати фактори, які впливають на купівлю чи відсутність купівлі цільового атрибуту (в даному випадку **Toothpaste**). Подібні правила можуть бути корисними для створення рекомендаційних систем чи оптимізації маркетингових стратегій.

3)

Сформував власний набір транзакцій покупок:

Receipt#	Bread	Honey	Bacon	Toothpaste	Banana	Apple	Hazelnut	Cheese	Meat	Carrot	Cucumber	Onion	Milk	Butter	ShavingFoam	Salt	Flour	HeavyCream	Egg	Olive	Shampoo	Sugar
1	y	n	y	n	y	y	y	n	n	y	n	n	n	n	n	n	n	y	y	n	n	y
2	y	y	y	n	y	y	y	n	n	n	y	n	y	y	n	n	y	n	n	y	y	n
3	n	y	y	y	y	y	y	y	y	n	y	y	y	n	y	y	y	y	n	n	n	y
4	y	y	n	y	n	y	n	n	n	n	y	y	y	n	n	n	y	n	y	y	y	n
5	n	y	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
6	n	y	n	y	n	n	y	n	n	n	n	y	n	n	y	n	n	n	n	n	n	y
7	n	n	y	n	y	y	n	n	n	y	n	n	n	n	n	y	n	y	n	n	n	n
8	n	n	y	y	y	n	y	n	n	n	y	y	n	n	n	y	n	n	y	n	n	n
9	n	y	y	n	y	y	y	y	y	y	n	y	n	y	y	n	n	y	y	y	n	n
10	n	n	n	n	n	n	n	n	n	n	n	y	y	n	n	y	y	y	n	n	y	n
11	n	n	y	y	n	y	y	n	n	y	n	n	n	y	n	n	y	y	y	y	n	n
12	n	y	y	y	n	n	n	n	y	n	n	y	n	y	n	n	y	y	n	n	n	y
13	y	y	y	y	y	y	n	n	y	y	n	y	y	n	y	n	n	n	y	y	y	y
14	y	n	y	y	y	n	y	y	n	y	n	n	n	n	y	y	n	y	y	y	n	y
15	n	n	n	y	y	n	n	n	n	y	y	n	n	n	y	n	y	n	y	y	y	y
16	n	n	y	n	n	n	n	n	n	n	n	n	y	n	n	n	y	n	n	n	y	n
17	n	n	n	y	n	n	n	n	n	n	y	n	n	n	n	n	y	n	n	n	n	n
18	n	n	n	n	n	n	n	n	n	n	n	y	n	n	y	y	n	n	n	y	n	n
19	y	y	y	y	n	n	y	y	y	y	n	y	n	y	y	y	n	n	n	n	n	y
20	y	n	y	y	y	n	y	y	n	y	n	y	n	y	y	y	y	n	n	y	n	y
21	n	n	n	n	n	n	n	y	n	n	n	n	n	n	y	n	n	n	n	n	y	y

Знайшов двоелементні групи частих наборів:

		Bread	Honey	Bacon	Toothpaste	Banana	Apple	Hazelnut	Cheese	Meat	Carrot	Cucumber	Onion	Milk	Butter	ShavingFoam	Salt	Flour	HeavyCream	Egg	Olive	Shampoo	Sugar
Support	Bread														0,196; 0		0,207;						
Confidence	0,1 Honey								0,203; 0											0,209;		0,188; 0,51	
	Bacon				0,241; 0,5			0,222; 0,5; 0,224; 0	0,216; 0					0,188; 0	0,207; 0	0,216; 0,532				0,203;			0,188; 0,512
	0,5 Toothpaste												0,194; 0										
	Banana	0,211; 0,4	0,213; 0	0,241; 0	0,203; 0,528		0,220; 0	0,222; 0,5; 0,224; 0	0		0,220; 0	0,196; 0,51	0,209; 0			0,220; 0,543				0,213; 0,205; 0			
	Apple																						
	Hazelnut				0,222; 0,203; 0,528										0,198; 0			0,200; 0	0,209; 0,503				
	Cheese	0,216; 0,4	0,218; 0	0,224; 0	0,198; 0,517	0,224; 0,5				0,205; 0	0,218; 0	0,192; 0,50	0,192; 0		0,216; 0	0,213; 0,527	0,200;			0,222;	0,188; 0,51	0,188; 0,512	
	Meat			0,216; 0																			0,188; 0,512
	Carrot				0,205; 0,534																0,203;		
	Cucumber																						
	Onion				0,194; 0,506																		
	Milk																						
	Butter																						
	ShavingFoam				0,216; 0									0,190; 0		0,190; 0							
	Salt	0,207; 0,4			0,194; 0,506																		0,188; 0,512
	Flour																						
	HeavyCream				0,192; 0,500					0,194; 0					0,207; 0			0,207;					
Egg		0,209; 0,4						0,222; 0															
Olive																							
Shampoo																							
Sugar																							

Висновок: У цій лабораторній роботі досліджено алгоритми Apriori та FP-Growth для пошуку асоціативних правил у наборах даних. У Weka було знайдено правила з високою достовірністю та підтримкою. Алгоритм Аргіогі виявив часті стабільні правила, тоді як FP-Growth працював швидше й знайшов рідкісні комбінації. В Excel створено набір транзакцій і знайдено двоелементні часті набори. Отримані результати можна застосувати для оптимізації викладки товарів та персоналізованих рекомендацій у супермаркетах.