وزارة التعليم العالي

جامعة حمص

الكلية التطبيقية

قسم تقنيات حاسوب

السنة الثالثة

قواعد معطيات٣

القسم العملي

المحاضرة الرابعة

اعداد المدرسين:

م. زينب مراد **م. بتول الليوس**

Association in DataMining

- It is making a simple correlation between two or more items, often of the same type to identify patterns

  "For example, when tracking people's buying habits, you might identify that a customer always buys cream when they buy strawberries, and therefore suggest that the next time that they buy strawberries they might also want to buy cream."

- **Discovering interesting relationships hidden in large datasets:**

example: Market basket analysis.

Similarly in an online shop, e.g. Amazon

transactions

items

Customer transactions

Tid	Transaction items
1	Butter, Bread, Milk, Sugar
2	Butter, Flour, Milk, Sugar
3	Butter, Eggs, Milk, Salt
4	Eggs
5	Butter, Flour, Milk, Salt, Sugar

- Given a set of customer transactions (shopping baskets), find:

  items that occur frequently together → **rules** that will predict the occurrence of an item based on the occurrences of other items.

- **Applications:** plan marketing or advertising strategies E.g., items that are frequently **purchased together** can be placed **in proximity** or placed **at opposite ends** .

Frequent Patterns الأنماط المتكررة

- **Patterns occur frequently in a target data set**
  - **Frequent itemset:** a **set of items** that often appear together in a **transactional dataset**.
    - ✓ E.g., What products are often purchased together?—milk and bread.
  - **(Frequent) Sequential pattern:** a **sequence** occurs frequently in a **sequence dataset**.
    - ✓ E.g., for a customer, What are the subsequent purchases after buying a new laptop?
  - **(Frequent) Structured pattern:** a **substructure** occurs frequently, where a substructure can refer to different structural forms e.g., subgraphs, subtrees, or sublattices.

An itemset X is a frequent itemset, If the support of X satisfies (no less than) a prespecified minimum support threshold (min_sup).

هي مجموعات من العناصر أو الأحداث التي تظهر بشكل متكرر في قاعدة البيانات, أي ان قيمة الدعم لها (Support)

تكون أعلى من الحد الأدنى المطلوب (Minimum Support Threshold)

اذا لم يتحقق الشرط فهي أنماط غير متكررة

transactions

item

Customer transactions

Tid	Transaction items
1	Butter, Bread, Milk, Sugar
2	Butter, Flour, Milk, Sugar
3	Butter, Eggs, Milk, Salt
4	Eggs
5	Butter, Flour, Milk, Salt, Sugar

- **items I**: the set of all items $I = \{I_1, I_2, \dots, I_m\}$.
  - e.g., products in a supermarket or online shop.
- **Itemset X:** A subset of items $X \subseteq I$.
  - e.g., {Milk, Bread, Diaper}
- **k-itemset:** an itemset of size k.
  - e.g., {Butter, Diapers, Milk} is a **3-itemset**.

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Market-Basket transactions

- **Transaction Database D:** the set of all transactions $D = \{T_1, T_2, \dots, T_n\}$.
- **Transaction :** $T = (tid, X_T)$, tid is the transaction identifier and X_T is the itemset contained in T.
- **Convention:** Items in transactions or itemsets are lexicographically ordered
  - Itemset $X = \{x_1, x_2, \dots, x_k\}$, such as $x_1 \leq x_2, \dots \leq x_k$.

-كل مناقلة أو معاملة T هي مجموعة فرعية من مجموعة العناصر I بالتوافق مع كل معاملة، يوجد رقم تعريف فريد يتم تسجيله على أنه معاملة TID

-بالنسبة لمجموعة العناصر X ، قم بتعيين العدد(X⊆ T) ليكون عدد المعاملات التي تحتوي علىX

في مجموعة المعاملاتD ثم يكون دعم مجموعة العناصرX هو $\frac{\text{عدد السجلات التي تحتوي } X}{\text{العدد الكلي للسجلات}}$

- تشكل المعاملة بأكملها قاعدة بيانات المعاملات D, و | D | يساوي العدد الكلي للمعاملات في D

- An important property of an itemset X is its support:
  - **Absolute support ($support\_count$)**: number of transactions containing the itemset.
    - ✓ E.g., $support\_count$({Bread, Milk, Diaper}) = 2
  - **Relative support (S)**: fraction of transactions containing the itemset.

$$s(x) = \frac{\text{support\_count(X)}}{|D|}$$

  - ✓ E.g., S({Bread, Milk, Diaper}) = 2/5

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Market-Basket transactions

Association Rules – Support

Support: Determines how often a rule is applicable to a given dataset.

الدعم هو مقياس يعبّر عن مدى تكرار ظهور عنصر أو مجموعة عناصر(Items) داخل قاعدة البيانات

نسبة عدد المعاملات (Transactions) التي تحتوي على العناصر Y و X معًا إلى العدد الكلي للمعاملات

$$S\ (x \Rightarrow y) = \frac{support\_count\ (x \cap y)}{|D|}$$

- **Example:**

$$S(\{Milk, diaper\} \Rightarrow \{Butter\}) = \frac{\text{support\_count}(\{Milk, Diaper, Butter\})}{|D|} = \frac{2}{5} = 0.4$$

  - 40% of all the transactions under analysis show that $Milk$, $Diaper$ and $Butter$ are purchased together.

مثال:

لنفترض أن لدينا 100 فاتورة مبيعات، و منها 20 تحتوي على (خبز و حليب) معا"

Support(bread,milk) = $\frac{20}{100}$ = 0.20 = 20%

أي أن ٢٠٪ من الفواتير تحتوي على هذا الزوج من المنتجات معا"

ملاحظة : الحد الأدنى للدعم هو الحد الأدنى للدعم لمجموعة العناصر ، والمشار إليه باسم min_sup

- تسمى مجموعة العناصر التي لا تقل درجة دعمها عنmin _sup مجموعة متكررة frequent patterns

أي (مرتفع Support)أنماط تظهر كثيرًا

- اذا لم يتحقق الشرط فهي مجموعة غير متكررة Infrequent patterns

(منخفض Support)أنماط نادرة الظهور

Association Rules – Confidence

Confidence: Measures the Reliability of the inference made by a rule.

يقيس احتمال أن تحتوي المعاملة على Yعند وجود X

$$C(x \Rightarrow y) = \frac{\text{support\_count}(x \cap y)}{\text{support\_count}(x)}$$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Market-Basket transactions

- **Example:**

$$C(\{Milk, diaper\} \Rightarrow \{Butter\}) = \frac{support\_count(\{Milk, Diaper, Butter\})}{support\_count(\{Milk, Diaper\})} = \frac{2}{3} = 0.67$$

  - 67% of the customers who purchased Milk and diaper also bought Butter.

Association Rules – Lift

Lift: The ratio of observed support to that expected if X and Y were independent.

النسبة بين الدعم المُلاحظ والدعم المتوقع في حال كان Y و X مستقلّين

$$L(x \Rightarrow y) = \frac{Confidence(x \cap y)}{support(y)}$$

$$L(x \Rightarrow y) = \frac{Support\_count(x \cap y)}{support\_count(x) \times support\_count(y)}$$

- **Lift > 1 implies a positive association**
  items occur together more than expected.

أي ان وجود عنصر يزيد من احتمال ظهور عنصر اخر, وجود علاقة إيجابية

- **Lift = 1 implies independence.** العنصرين مستقلين تماما"
- **Lift < 1 implies a negative association**

وجود عنصر يقلل من احتمال ظهور عنصر اخر, وجود علاقة سلبية أو عكسية

TID	Items
1	Bread, Milk
2	Bread, Diaper, Butter, Eggs
3	Milk, Diaper, Butter, Coke
4	Bread, Milk, Diaper, Butter
5	Bread, Milk, Diaper, Coke

Market-Basket transactions

Example:

$$Support(\{Milk \Rightarrow Diaper\}) = \frac{Support\_count(\{Milk\} \cap \{Diaper\})}{|D|} = \frac{3}{5} = 0.6$$

$$Support(\{Butter\}) = \frac{Support\_count(\{Butter\})}{|D|} = = \frac{3}{5} = 0.6$$

$$Support(\{Milk,Diaper\} \Rightarrow \{Butter\}) = \frac{Support\_count(\{Milk,Diaper\} \cap \{Butter\})}{|D|} = \frac{2}{5} = 0.4$$

$$Confidence(\{Milk,Diaper\} \Rightarrow \{Butter\}) = \frac{Support(\{Milk,Diaper\} \cap \{Butter\})}{support(\{Milk,Diaper\})} = \frac{0.4}{0.6} = 0.6667$$

- $Lift(\{Milk,Diaper\} \Rightarrow \{Butter\}) = \frac{Confidence(\{Milk,Diaper\} \cap \{Butter\})}{support(\{Butter\})} = \frac{0.67}{0.6} = 1.1 > 1$

  Or

- $Lift(\{Milk,Diaper\} \Rightarrow \{Butter\}) = \frac{support\_count(\{Milk,Diaper\} \cap \{Butter\})}{Support(\{Milk \Rightarrow Diaper\}) \times support(\{Butter\})} = \frac{0.4}{0.36} = 1.1 > 1$

possitive association هناك ارتباط إيجابي ضعيف بين العناصر الثلاثة

Frequent Itemset Mining Methods

- Several Strategies to reduce the **computational complexity** of FIM.
  - Reduce the number of **candidate itemsets (M)**.
  - Reduce the number of **transactions (N)**
  - Reduce the number of **comparisons (NM)**.

- Many algorithms:
  - Apriori Algorithm,
  - Dynamic Hash and Pruning (DHP),
  - Frequent Pattern-Growth Approach (FP-Growth),
  - H-Mine
  - …

Apriori Algorithm

- **Convention**: For efficient implementation, Apriori Algorithm assumes that items within a transaction or itemset are sorted in **lexicographic order**.
- **Notation:**
  - L_k: set of **frequent itemsets** of size k
  - C_k: set of **candidate itemsets** of size k

1. Association Rule Mining (ARM):

Given: a set of items (*I*), a transactional Database *D* over *I* and minimum thresholds *min_sup* & *min_conf*.

Our Goal: find all strong association rules in D:

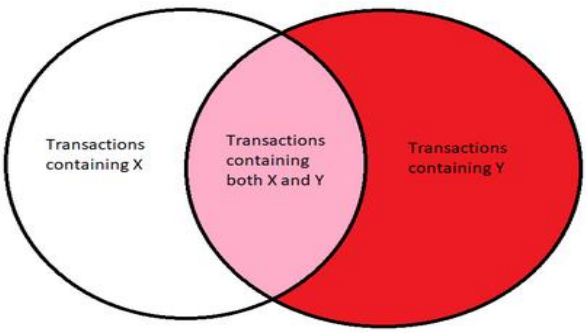
- 2-step method to extract the association rules:
  Step 1: Determine the frequent itemsets w.r.t. min support. [**FIM (Frequent Itemset Mining)** problem (Apriori Algorithm)]

  Apriori Algorithm overview:

  - ✓ Initially, scan transaction DB once to get frequent 1 itemset $\boldsymbol{L1}$.
  - ✓ Generate length (k+ 1) candidate itemsets $\boldsymbol{C_{k+1}}$ from length k frequent itemsets $\boldsymbol{L_k}$. *(Join Step)*
  - ✓ Evaluate whether the candidates $\boldsymbol{C_{k+1}}$ are really frequent, query the DB →frequent (k+1)-itemsets $\boldsymbol{L_{k+1}}$. *(prune Step)*
  - ✓ Terminate when no frequent or candidate set can be generated.

  Step 2: using the frequent itemsets found in the previous step, Generate the association rules w.r.t. min confidence.

  - ✓ For each frequent itemset $\boldsymbol{X}$, generate all nonempty subsets.
  - ✓ For every nonempty subset $\boldsymbol{Y}$ of $\boldsymbol{X}$, output the rule "$\boldsymbol{Y \Rightarrow (X - Y)}$", if $\frac{\boldsymbol{support\_count(X)}}{\boldsymbol{support\_count(Y)}} \geq \boldsymbol{min\_conf}$.



مبدأ خوارزمية Apriori

يعتمد اسمها على حقيقة أن الخوارزمية تستخدم خاصية الأولوية لخاصية العناصر المتكررة ، أي أن جميع المجموعات الفرعية غير الفارغة لمجموعة العناصر المتكررة يجب أن تكون متكررة أيضًا. تستخدم خوارزمية Aprioriطريقة تكرارية تسمى البحث تلو الأخرى,

حيث يتم استخدام مجموعةالعناصرk لاستكشاف مجموعة العناصر(k+1)

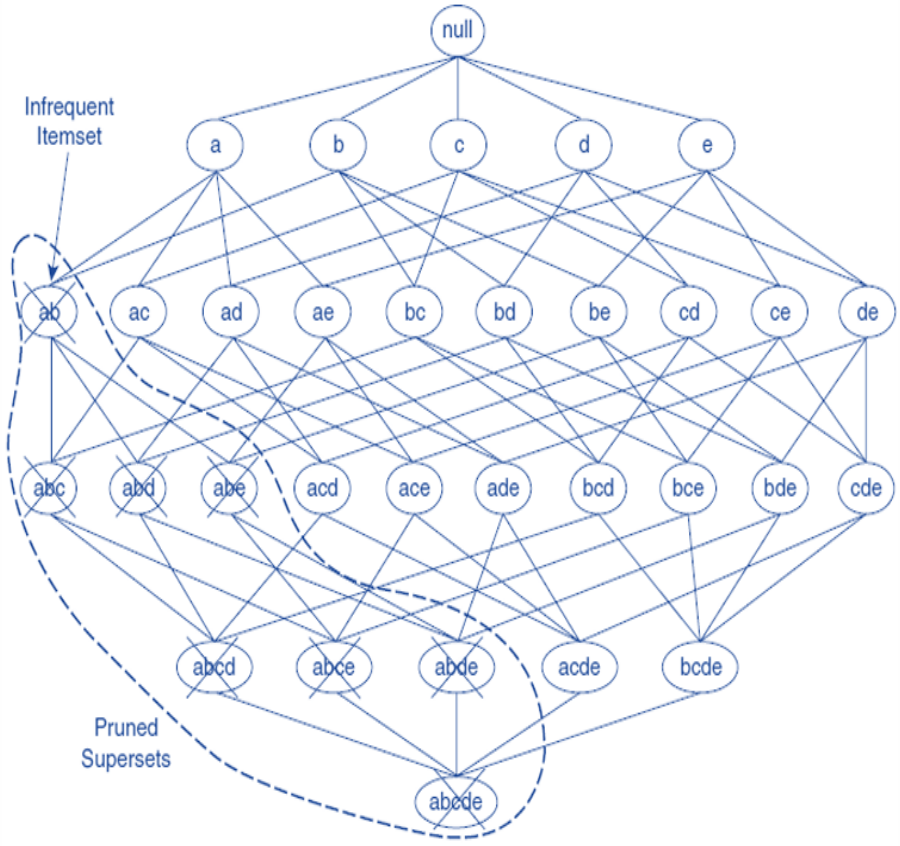
-أولاً ، قم بمسح قاعدة البيانات ، وتجميع عدد كل عنصر ، وجمع العناصر التي تلبي الحد ثم عناصر الأدنى من درجة الدعم للعثور على مجموعة مجموعات العناصر ١ المتكررة.

- يُشار إلى هذه المجموعةباسمL1 استخدمها للعثور على مجموعةL2 لمجموعتي عناصر متكررتين,واستخدم L2 للبحث عن L3 وهكذا

حتى لم يعد من الممكن العثور على مجموعة عناصر k المتكررة.

يتطلب كل Lk موجود مسح كامل قاعدة البيانات .

Pruning based on Apriori principle: If {a, b} is infrequent, then all supersets of {a, b} are infrequent.



itemset lattice for I = {a, b, c, d, e}

A lattice structure can be used to enumerate all the possible itemsets

Example -1-

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

Min_supp=2

How to generate candidates?

- Step 1: self-joining L_k
- Step 2: pruning

Example (step 1):

L_3={abc, abd, acd, ace, bcd}

- Self-joining: $L_3 \bowtie L_3$
  - abcd from abc and abd
  - acde from acd and ace
- Pruning:
  - acde is removed because ade is not in L_3
- C_4 = {abcd}

{D} infrequent:
Apriori property (any superset contains {D} will be infrequent) → not generated or tested as a candidate)

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

- Given the following transaction database and a min_Sup =4, find all frequent itemsets.
  - Report on C_k , L_k sets as well as on how L_k was derived from C_k(Apriori pruning or support count).

Database TDB

Tid	Items
1	A, B, C, D, E
2	C, D, E, F, G
3	A, B, C, D
4	B, C, D, E
5	A, D, E, F
6	A, B, C, E
7	B, C, E, F
8	A, B, G
9	A, B, C, E, F
10	A, C, D, E

minSupport s = 4

Database TDB

Tid	Items
1	A, B, C, D, E
2	C, D, E, F, G
3	A, B, C, D
4	B, C, D, E
5	A, D, E, F
6	A, B, C, E
7	B, C, E, F
8	A, B, G
9	A, B, C, E, F
10	A, C, D, E

minSupport = 4

1st scan →

C_1

Itemset	sup
{A}	7
{B}	7
{C}	8
{D}	6
{E}	8
{F}	4
{G}	2

L_1

Itemset	sup
{A}	7
{B}	7
{C}	8
{D}	6
{E}	8
{F}	4

C_2

Itemset
{A, B}
{A, C}
{A, D}
{A, E}
{A, F}
{B, C}
{B, D}
{B, E}
{B, F}
{C, D}
{C, E}
{C, F}
{D, E}
{D, F}
{E, F}

2nd scan →

Itemset	sup
{A, B}	5
{A, C}	5
{A, D}	4
{A, E}	5
{A, F}	2
{B, C}	6
{B, D}	3
{B, E}	5
{B, F}	2
{C, D}	5
{C, E}	7
{C, F}	3
{D, E}	5
{D, F}	2
{E, F}	4

→

L_2

Itemset	sup
{A, B}	5
{A, C}	5
{A, D}	4
{A, E}	5
{B, C}	6
{B, E}	5
{C, D}	5
{C, E}	7
{D, E}	5
{E, F}	4

C_3

Itemset
{A, B, C}
{A, B, D}
{A, B, E}
{A, C, D}
{A, C, E}
{A, D, E}
{B, C, E}
{C, D, E}

{B, D} is non frequent -> pruned by Apriori property

3rd scan

Itemset	sup
{A, B, C}	4
{A, B, E}	3
{A, C, D}	3
{A, C, E}	4
{A, D, E}	3
{B, C, E}	5
{C, D, E}	4

L_3

Itemset	sup
{A, B, C}	4
{A, C, E}	4
{B, C, E}	5
{C, D, E}	4

- Green rows: prune by Apriori property
- Red rows: prune by minSupport threshold

Application Example:

1. First Step: install Required Libraries:

Before we start, ensure that you are install *apyori* library using this command:

```
pip install apyori  #use pip3 instead of pip if you using python3 version
```

2. import libraries and load our dataset:

We'll import pandas in order to read and view our transactional dataset, and import Apriori function which implement previous two steps.

1) Import required libraries:

```
import pandas as pd
from apyori import apriori
import numpy as np
```

2) Load our dataset using Pandas:

#header parameter to read first line as transaction not attributes names

```
store_data_df = pd.read_csv('store_data.csv', header=None)
store_data_df
```

Our dataset looks like the following:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
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	sp
1	burgers	meatballs	eggs	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	
3	turkey	avocado	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	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7496	butter	light mayo	fresh bread	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7497	burgers	frozen vegetables	eggs	french fries	magazines	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7498	chicken	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7499	escalope	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
7500	eggs	frozen smoothie	yogurt cake	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

501 rows × 20 columns

3. Preprocessing transactional dataset:

The Apriori library requires our dataset to be in the form of a list of lists, where the whole dataset is a big list and each transaction in the dataset is an inner list within the outer big list.

To convert our pandas dataframe into a list of lists, execute the following script:

```
#transfer dataframe to list of lists
transactions = []
for i in range(0, store_data_df.shape[0]):          #walk on rows
    temp = []
    for j in range(0, store_data_df.shape[1]):       #walk on columns
        if not store_data_df.loc[i,j] is np.nan:      #test value[i,j] is not 'NaN'
            temp.append(store_data_df.loc[i,j])
    transactions.append(temp)
```

Note: We need to take care to remove the 'NaN' items, since if they remain in the dataset, 'NaN' also appears in the association rules.

4. Applying Apriori Algorithm:

The Apriori function parameters described in the following table:

#	Parameter Name	Type	Explanation
1	Transactions	List	Transaction dataset, in our example is the list of lists derived from our dataframe.
2	min_support	Float	Minimum support threshold: to select the itemsets with support values greater than the value specified by this parameter.
3	min_confidence	Float	Minimum confidence threshold: To filter those rules that have confidence greater than the value specified by this parameter.
4	min_lift	Float	Minimum lift value: if the value of lift measure is greater than 1, this mean that items in the antecedent and consequent of the rule are positively correlated.

Let's pass the following values:

- ✓ We want rules for only those items that are purchased at least 5 times a day, or 7 x 5 = 35 times in one week, since our dataset is for a one-week time period. The support for those items can be calculated as 35/7500 = 0.0045.
- ✓ The minimum confidence for the rules is 0.01 or 1%.
- ✓ The lift value is 3.

```
#applying Apriori Algorithm
rules = apriori(transactions= transactions, min_support=0.0045, min_confidence=0.01, min_lift=3)
association_results = list(rules)
print("total Rules after run the Algorithm:",len(association_results))
```

total Rules after run the Algorithm: 25

5. Explain Output:

To make our output clear, we're going to print the rules and their support, confidence and lift:

```
for i in range(len(association_results)):
    print(association_results[i])
```

and our result is:

```
RelationRecord(items=frozenset({'whole wheat pasta', 'olive oil'}), support=0.007998933475536596, ordered_statistics=[OrderedS
tatistic(items_base=frozenset({'olive oil'}), items_add=frozenset({'whole wheat pasta'}), confidence=0.12145748987854252, lift
=4.1224100976422955), OrderedStatistic(items_base=frozenset({'whole wheat pasta'}), items_add=frozenset({'olive oil'}), confid
ence=0.2714932126696833, lift=4.122410097642296)])
RelationRecord(items=frozenset({'pasta', 'shrimp'}), support=0.005065991201173177, ordered_statistics=[OrderedStatistic(items_
base=frozenset({'pasta'}), items_add=frozenset({'shrimp'}), confidence=0.3220338983050847, lift=4.506672147735896), OrderedSta
tistic(items_base=frozenset({'shrimp'}), items_add=frozenset({'pasta'}), confidence=0.0708955223880597, lift=4.50667214773589
6)])
RelationRecord(items=frozenset({'frozen vegetables', 'shrimp', 'chocolate'}), support=0.005332622317024397, ordered_statistics
=[OrderedStatistic(items_base=frozenset({'frozen vegetables'}), items_add=frozenset({'shrimp', 'chocolate'}), confidence=0.055
944055944055944, lift=3.1084175084175087), OrderedStatistic(items_base=frozenset({'shrimp'}), items_add=frozenset({'frozen veg
etables', 'chocolate'}), confidence=0.07462686567164178, lift=3.2545123221103784), OrderedStatistic(items_base=frozenset({'fro
zen vegetables', 'chocolate'}), items_add=frozenset({'shrimp'}), confidence=0.23255813953488375, lift=3.2545123221103784), Ord
eredStatistic(items_base=frozenset({'shrimp', 'chocolate'}), items_add=frozenset({'frozen vegetables'}), confidence=0.29629629
629629634, lift=3.1084175084175087)])
```

References:

1- Pandas docs [https://pandas.pydata.org/docs/].
2- NumPy docs [https://numpy.org/doc/].
3- Matplotlib docs [https://matplotlib.org/3.3.3/contents.html].
4- SciKit-learn docs [https://scikit-learn.org/0.21/documentation.html].
5- Apyori webpage [https://pypi.org/project/apyori/].
6- https://medium.com/@sumedh0803/extracting-association-rules-from-grocery-store-data-ad776ae2d34e