Fouille de Données

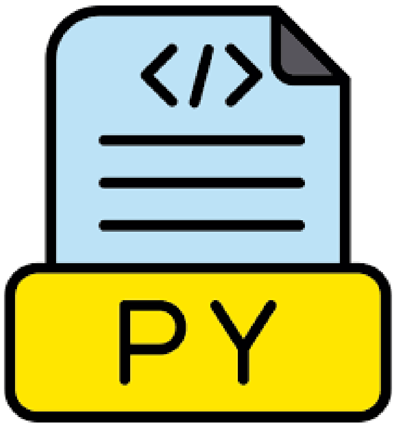
Data Mining

Recherche des Motifs Fréquents

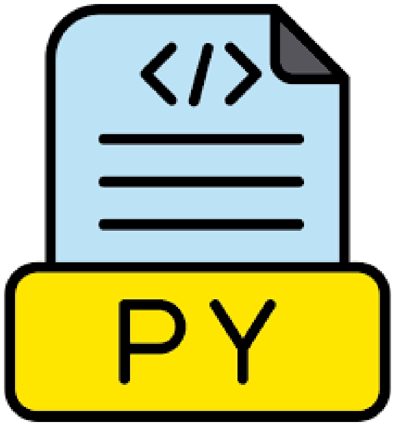
et Extraction des Règles d'Association

Motifs Fréquents et Règles d'association - Apriori

- **Série TP 1 - Python**



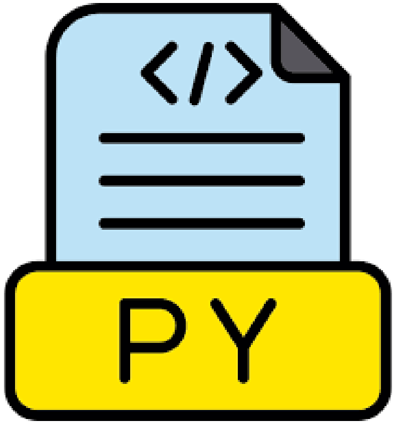
Partie 1 - Découverte



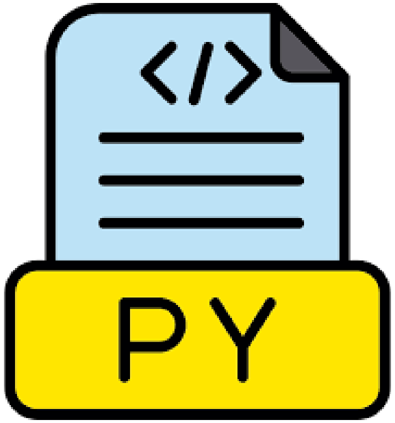
Partie 2 - Exercices

Motifs Fréquents et Règles d'association - Apriori

- **Série TP 2 - Apriori**



Partie 1 - Découverte



Partie 2 - Exercices

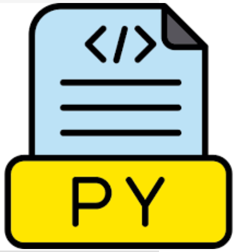
Motifs Fréquents et Règles d’association - Apriori

Install & Import - apriori_python library

In [1]:

```
!pip install apriori_python
```

```
Collecting apriori_python
  Downloading apriori_python-1.0.4-py3-none-any.whl (5.0 kB)
Installing collected packages: apriori-python
Successfully installed apriori-python-1.0.4
```



In [1]:

```
from apriori_python import apriori
```

https://github.com/chonyy/apriori_python

https://github.com/chonyy/apriori_python/blob/master/apriori_python/apriori.py

Motifs Fréquents et Règles d'association - Apriori

Dataset 1 - Execrice TD 1

```
transactions_list = [

]
```

Motifs Fréquents et Règles d'association - Apriori

Dataset 1 - Execrice TD 1

```
transactions_list = [
        ['P1', 'P2', 'P3'],
        ['P1', 'P3'],
        ['P1', 'P2', 'P3'],
        ['P1', 'P3'],
        ['P2', 'P3'],
        ['P4'],
]
```

Motifs Fréquents et Règles d'association - Apriori

```
freq_itemsets, rules = apriori(transactions_list, minSup=0.333, minConf=0.55)
```

Motifs Fréquents et Règles d’association - Apriori

```python
freq_itemsets, rules = apriori(transactions_list, minSup=0.333, minConf=0.55)
```

```python
freq_itemsets
```

```
{1: {frozenset({'P3'}), frozenset({'P2'}), frozenset({'P1'})},
 2: {frozenset({'P2', 'P3'}),
  frozenset({'P1', 'P3'}),
  frozenset({'P1', 'P2'})},
 3: {frozenset({'P1', 'P2', 'P3'})}}
```

Motifs Fréquents et Règles d’association - Apriori

```
freq_itemsets, rules = apriori(transactions_list, minSup=0.333, minConf=0.55)
```

```
freq_itemsets
```

```
{1: {frozenset({'P3'}), frozenset({'P2'}), frozenset({'P1'})},
 2: {frozenset({'P2', 'P3'}),
  frozenset({'P1', 'P3'}),
  frozenset({'P1', 'P2'})},
 3: {frozenset({'P1', 'P2', 'P3'})}}
```

Motifs Fréquents et Règles d'association - Apriori

```
freq_itemsets, rules = apriori(transactions_list, minSup=0.333, minConf=0.55)
```

```
rules
```

```
[[{'P3'}, {'P2'}, 0.6],
 [{'P2'}, {'P1'}, 0.6666666666666666],
 [{'P2'}, {'P1', 'P3'}, 0.6666666666666666],
 [{'P2', 'P3'}, {'P1'}, 0.6666666666666666],
 [{'P3'}, {'P1'}, 0.8],
 [{'P2'}, {'P3'}, 1.0],
 [{'P1'}, {'P3'}, 1.0],
 [{'P1', 'P2'}, {'P3'}, 1.0]]
```

Motifs Fréquents et Règles d’association - Apriori

```
freq_itemsets, rules = apriori(transactions_list, minSup=0.333, minConf=0.55)
```

```
rules
```

```
[[{'P3'}, {'P2'}, 0.6],
 [{'P2'}, {'P1'}, 0.6666666666666666],
 [{'P2'}, {'P1', 'P3'}, 0.6666666666666666],
 [{'P2', 'P3'}, {'P1'}, 0.6666666666666666],
 [{'P3'}, {'P1'}, 0.8],
 [{'P2'}, {'P3'}, 1.0],
 [{'P1'}, {'P3'}, 1.0],
 [{'P1', 'P2'}, {'P3'}, 1.0]]
```

Motifs Fréquents et Règles d’association - Apriori

Larger CSV Dataset - Market Basket Analysis

Import libraries

```
import io
import pandas as pd
```

pandas - Python Data Analysis Library

Motifs Fréquents et Règles d’association - Apriori

Larger CSV Dataset - Market Basket Analysis

```python
df = pd.read_csv('Market_Basket_Optimisation.csv', header = None)
```

A1 | f_x shrimp,almonds,avocado,vegetables mix,green grapes,whole weat flour,yams,co

	A	B	C	D	E	F	G	H	I
1	shrimp,almonds,avocado,vegetables mix,green grapes,whole weat flour,yams,cottage cheese,energy drink,tomato juice								
2	burgers,meatballs,eggs								
3	chutney								
4	turkey,avocado								
5	mineral water,milk,energy bar,whole wheat rice,green tea								
6	low fat yogurt								
7	whole wheat pasta,french fries								
8	soup,light cream,shallot								
9	frozen vegetables,spaghetti,green tea								
10	french fries								
11	eggs,pet food								
12	cookies								
13	turkey,burgers,mineral water,eggs,cooking oil								
14	spaghetti,champagne,cookies								
15	mineral water,salmon								
16	mineral water								
17	shrimp,chocolate,chicken,honey,oil,cooking oil,low fat yogurt								
18	turkey,eggs								
19	turkey,fresh tuna,tomatoes,spaghetti,mineral water,black tea,salmon,eggs,chicken,extra dark chocolate								
20	meatballs,milk,honey,french fries,protein bar								
21	red wine,shrimp,pasta,pepper,eggs,chocolate,shampoo								

Market_Basket_Optimisation

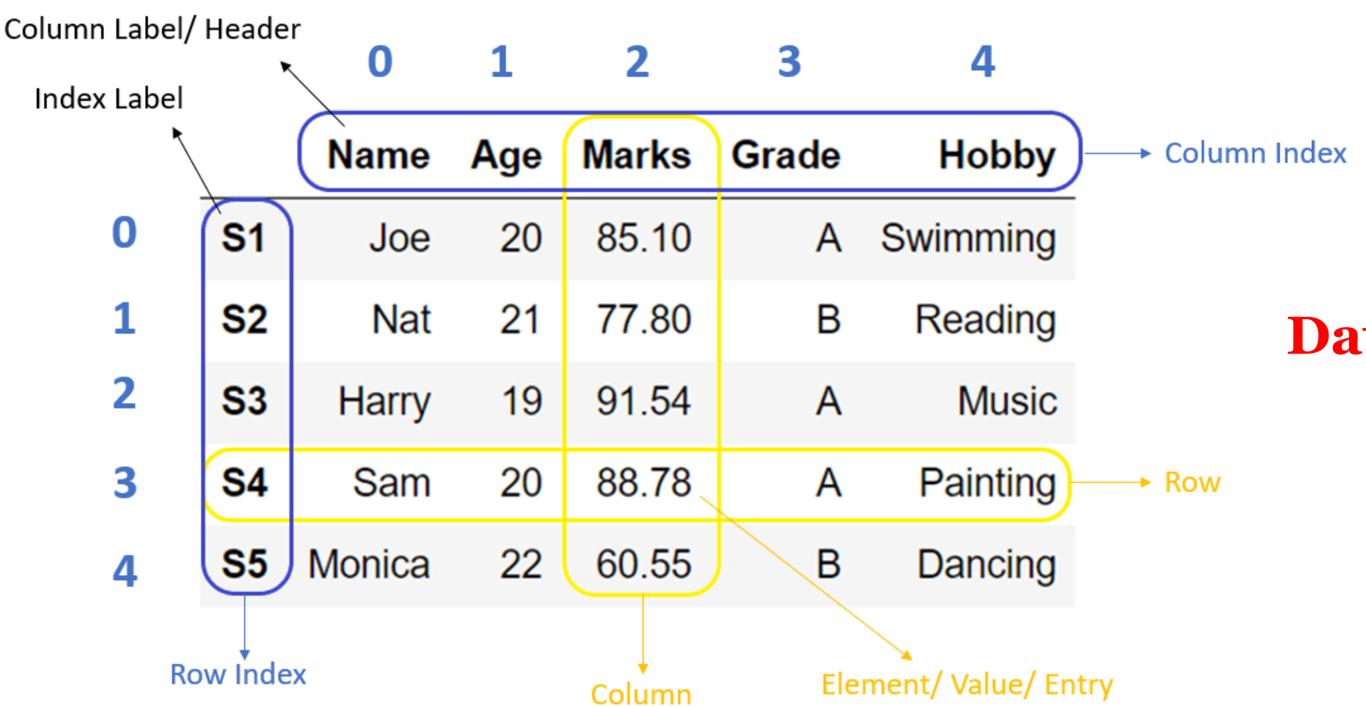
Nom

- Advertising.csv
- apriori_python-1.0.4-py3-none-any.whl
- iris.csv
- Market_Basket_Optimisation.csv
- Online_Retail.xlsx
- Online_Retail_Cleaned.csv
- tennis.csv
- weather_numeric.csv

Motifs Fréquents et Règles d’association - Apriori

Larger CSV Dataset - Market Basket Analysis

```python
df = pd.read_csv('Market_Basket_Optimisation.csv', header = None)
```



DataFrame

Motifs Fréquents et Règles d'association - Apriori

Larger CSV Dataset - Market Basket Analysis **DataFrame**

Basic Information

```
>>> df.shape #(rows,columns)
>>> df.index #Describe index
>>> df.columns #Describe DataFrame columns
>>> df.info() #Info on DataFrame
>>> df.count() #Number of non-NA values
```

Summary

```
>>> df.sum() #Sum of values
>>> df.cumsum() #Cummulative sum of values
>>> df.min()/df.max() #Minimum/maximum values
>>> df.idxmin()/df.idxmax() #Minimum/Maximum index value
>>> df.describe() #Summary statistics
>>> df.mean() #Mean of values
```

Motifs Fréquents et Règles d’association - Apriori

Larger CSV Dataset - Market Basket Analysis

Printing first 5 rows of the Dataset

```
df.head(5)
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
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	sal
1	burgers	meatballs	eggs	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	
3	turkey	avocado	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	

Data shape (rows, cols)

```
df.shape
```

(7501, 20)

Motifs Fréquents et Règles d'association - Apriori

Larger CSV Dataset - Market Basket Analysis

```
df[0]
```

```
0                shrimp
1               burgers
2               chutney
3                turkey
4         mineral water
              ...
7496             butter
7497            burgers
7498            chicken
7499           escalope
7500               eggs
Name: 0, Length: 7501, dtype: object
```

df['column_name']

df.column_name

Motifs Fréquents et Règles d'association - Apriori

Larger CSV Dataset - Market Basket Analysis

```
df.loc[0]
```

```
0                  shrimp
1                 almonds
2                 avocado
3          vegetables mix
4            green grapes
5        whole weat flour
6                    yams
7          cottage cheese
8            energy drink
9            tomato juice
10         low fat yogurt
11              green tea
12                  honey
13                  salad
14          mineral water
15                 salmon
16       antioxydant juice
17        frozen smoothie
18                spinach
19              olive oil
Name: 0, dtype: object
```

df.loc[row] (label-based indexing)

df.iloc[row] (integer-based indexing)

df.iloc[] (integer-based indexing)

Motifs Fréquents et Règles d’association - Apriori

Larger CSV Dataset - Market Basket Analysis

```
df.values[0]
```

```
array(['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'], dtype=object)
```

df.values[row]

Motifs Fréquents et Règles d'association - Apriori

Larger CSV Dataset - Market Basket Analysis

```
# Access element at row 0, column 1
df.at[0, 1]
```

```
'almonds'
```

```
# OR
df.values[0, 1]
```

```
'almonds'
```

```
df.iat[0, 1]
```

```
'almonds'
```

Motifs Fréquents et Règles d'association - Apriori

Larger CSV Dataset - Market Basket Analysis

Build the Apriori model

```
freqItemSet, rules = apriori(transacts, minSup=0.01, minConf=0.2)
```

```
freqItemSet
```

...

```
rules
```

Motifs Fréquents et Règles d'association - Apriori

Larger CSV Dataset - Market Basket Analysis

Convert Pandas DataFrame df into a list of lists

```
transacts = []

for i in range(0, 7501):
    row = []
    for j in range(0, 20):
        if str(df.values[i, j]) != 'nan':
            row.append(str(df.values[i, j]))
    transacts.append(row)
```

Motifs Fréquents et Règles d’association - Apriori

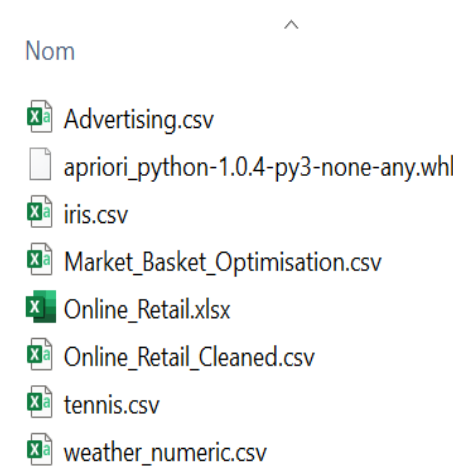
Numerical Dataset - weather.numeric dataset

Load Dataset

```
dataset = pd.read_csv('weather_numeric.csv')
```

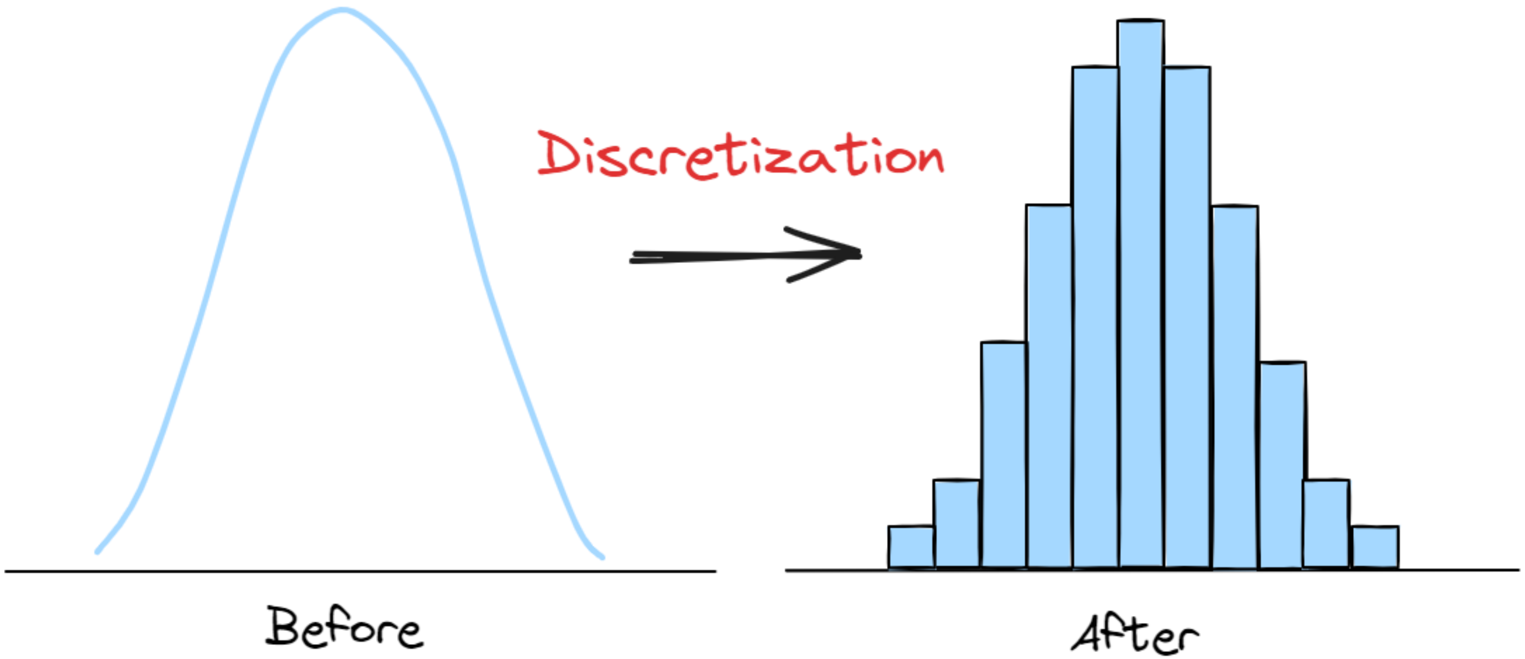
```
dataset.head()
```

	outlook	temperature	humidity	windy	play
0	sunny	85	85	False	no
1	sunny	80	90	True	no
2	overcast	83	86	False	yes
3	rainy	70	96	False	yes
4	rainy	68	80	False	yes



Motifs Fréquents et Règles d’association - Apriori

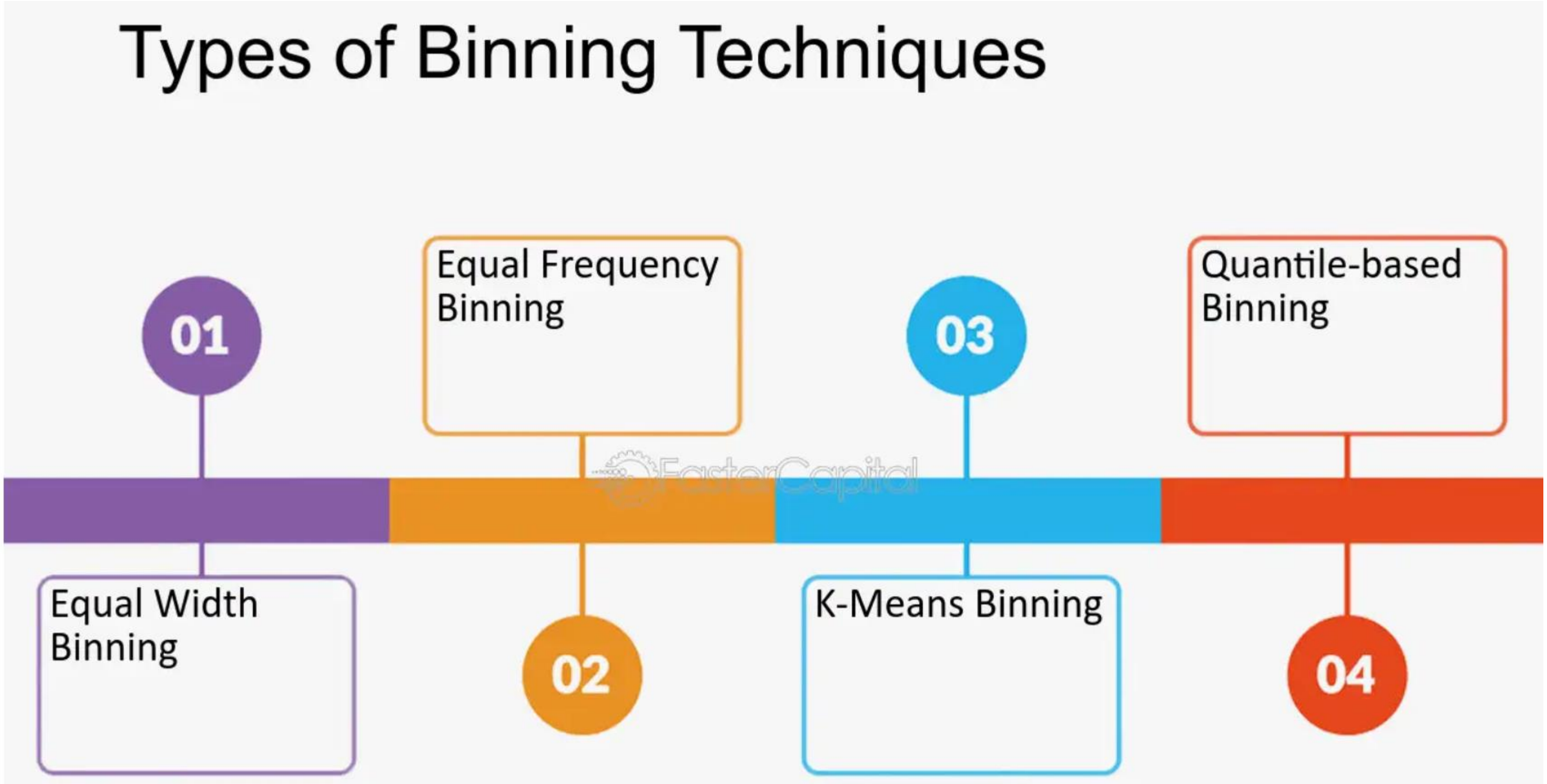
Numerical Dataset - weather.numeric dataset



The process of converting a **numerical values** into **categorical values**

Motifs Fréquents et Règles d’association - Apriori

Numerical Dataset - weather.numeric dataset



Motifs Fréquents et Règles d'association - Apriori

Numerical Dataset - weather.numeric dataset

Plusieurs **stratégies** possibles :

- Stratégie uniforme (Equal Width Binning) : En partageant l'intervalle des valeurs possibles de l'attribut en intervalles (bins) de taille égale.
- Stratégie quantile (Quantile Binning) : En le partageant en intervalles contenant le même nombre d'éléments.
- Stratégie k-means (Clustering Binning): Classes basées sur un Algorithme de k-means.
- Stratégie personnalisée : En fixant manuellement le nombre d'intervalles .
- Etc.

Motifs Fréquents et Règles d'association - Apriori

Numerical Dataset - weather.numeric dataset

- **Data :** 0, 4, 12, 16, 16, 18, 24, 26, 28
- **Equal width**
  - Bin 1: 0, 4 — [-,10)
  - Bin 2: 12, 16, 16, 18 — [10,20)
  - Bin 3: 24, 26, 28 — [20,+)
- **Equal frequency**
  - Bin 1: 0, 4, 12 — [-, 14)
  - Bin 2: 16, 16, 18 — [14, 21)
  - Bin 3: 24, 26, 28 — [21,+)

Motifs Fréquents et Règles d’association - Apriori

Numerical Dataset - weather.numeric dataset

Example (Quartile Binning - 4 bins) :

Dataset : [5, 12, 19, 22, 25, 35, 40, 50, 65, 80]

We divide it into **4 quantiles** (each containing 25% of the data):

- **Q1 (0-25%)**: [5, 12, 19]
- **Q2 (25-50%)**: [22, 25, 35]
- **Q3 (50-75%)**: [40, 50, 65]
- **Q4 (75-100%)**: [80]

Motifs Fréquents et Règles d’association - Apriori

Numerical Dataset - weather.numeric dataset

Transformation des données - Discrétisation Personnalisée

Temperature qui doivent être discrétisées selon les modalités suivantes :

- **(min-70.5]** - **Cool** : valeurs inférieures ou égales à 70,5.
- **(70.5–77.5] - Temperate** : valeurs entre 70,5 et 77,5 incluse.
- **(77.5–max[** - **Hot** : valeurs supérieures à 77,5.

Humidity doivent être discrétisées selon les modalités suivantes :

- (min-77.5] - **low** : valeurs inférieures ou égales à 77,5.
- (77.5-88] - **medium** : valeurs entre 77,5 et 88 incluse.
- (88-max] - **high** : valeurs supérieures à 88.

Motifs Fréquents et Règles d'association - Apriori

Numerical Dataset - weather.numeric dataset

```
temps = ['cool', 'temperate', 'hot']

dataset['temp_discret'] = pd.cut(x=dataset['temperature'],
                                 bins=[63, 70.5, 77.5, 85],
                                 labels=temps)
```

```
humdits = ['low', 'medium', 'high']

dataset['humdt_discret'] = pd.cut(x=dataset['humidity'],
                                  bins=[64, 77.5, 88, 96],
                                  labels=humdits)
```

Motifs Fréquents et Règles d'association - Apriori

pandas.cut

pandas.cut(*x*, *bins*, *right*=True, *labels*=None, *retbins*=False, *precision*=3, *include_lowest*=False, *duplicates*='raise', *ordered*=True)

[source]

- Discretize into **three equal-sized** bins.

```
temps = ['cool', 'temperate', 'hot']

dataset['temp_discret'] = pd.cut(x=dataset['temperature'],
                                  bins=3,
                                  labels=temps)
```

```
Name: temperature, dtype: category
Categories (3, interval[float64, right]): [(63.979, 71.0] < (71.0, 78.0] < (78.0, 85.0]]
```

Motifs Fréquents et Règles d’association - Apriori

```
pd.cut(dataset['temperature'],
       bins=[63, 70.5, 77.5, 85],
       right=True)
```

```
0     (77.5, 85.0]
1     (77.5, 85.0]
2     (77.5, 85.0]
3     (63.0, 70.5]
4     (63.0, 70.5]
5     (63.0, 70.5]
6     (63.0, 70.5]
7     (70.5, 77.5]
8     (63.0, 70.5]
9     (70.5, 77.5]
10    (70.5, 77.5]
11    (70.5, 77.5]
12    (77.5, 85.0]
13    (70.5, 77.5]
Name: temperature, dtype: category
Categories (3, interval[float64, right]): [(63.0, 70.5] < (70.5, 77.5] < (77.5, 85.0]]
```

[(63.0, 70.5] < (70.5, 77.5] < (77.5, 85.0]]

Motifs Fréquents et Règles d'association - Apriori

```
pd.cut(dataset['temperature'],
       bins=[63, 70.5, 77.5, 85],
      right=False)
```

```
0                NaN
1       [77.5, 85.0)
2       [77.5, 85.0)
3       [63.0, 70.5)
4       [63.0, 70.5)
5       [63.0, 70.5)
6       [63.0, 70.5)
7       [70.5, 77.5)
8       [63.0, 70.5)
9       [70.5, 77.5)
10      [70.5, 77.5)
11      [70.5, 77.5)
12      [77.5, 85.0)
13      [70.5, 77.5)
Name: temperature, dtype: category
Categories (3, interval[float64, left]): [[63.0, 70.5) < [70.5, 77.5) < [77.5, 85.0)]
```

[[63.0, 70.5) < [70.5, 77.5) < [77.5, 85.0)]

Motifs Fréquents et Règles d’association - Apriori

```
dataset
```

	outlook	temperature	humidity	windy	play	temp_discret	humdt_discret
0	sunny	85	85	False	no	hot	medium
1	sunny	80	90	True	no	hot	high
2	overcast	83	86	False	yes	hot	medium
3	rainy	70	96	False	yes	cool	high
4	rainy	68	80	False	yes	cool	medium
5	rainy	65	70	True	no	cool	low
6	overcast	64	65	True	yes	cool	low
7	sunny	72	95	False	no	temperate	high
8	sunny	69	70	False	yes	cool	low
9	rainy	75	80	False	yes	temperate	medium
10	sunny	75	70	True	yes	temperate	low

Motifs Fréquents et Règles d’association - Apriori

	outlook	windy	play	temp_discret	humdt_discret
0	sunny	False	no	hot	medium
1	sunny	True	no	hot	high
2	overcast	False	yes	hot	medium
3	rainy	False	yes	cool	high
4	rainy	False	yes	cool	medium
5	rainy	True	no	cool	low
6	overcast	True	yes	cool	low
7	sunny	False	no	temperate	high
8	sunny	False	yes	cool	low

```
dataset.drop('temperature', axis=1, inplace=True)

dataset.drop('humidity', axis=1, inplace=True)

dataset.head()
```

Motifs Fréquents et Règles d'association - Apriori

Numerical Dataset - weather.numeric dataset

```
dataset_list = []

for i in range(0, 14):
  dataset_list.append([str(dataset.values[i,j]) for j in range(0, 5)])

dataset_list[0:3]
```

```
[['sunny', 'False', 'no', 'hot', 'medium'],
 ['sunny', 'True', 'no', 'hot', 'high'],
 ['overcast', 'False', 'yes', 'hot', 'medium']]
```

```
freqItemSet, rules = apriori(dataset_list, minSup=0.2, minConf=0.5)
```

```
freqItemSet
```

...

```
rules
```

Motifs Fréquents et Règles d'association - Apriori

Numerical Dataset - weather.numeric dataset

KBinsDiscretizer

class **sklearn.preprocessing.KBinsDiscretizer**(*n_bins=5*, *, *encode='onehot'*, *strategy='quantile'*, *dtype=None*, *subsample=200000*, *random_state=None*) [source]

Bin continuous data into intervals.

strategy : *{'uniform', 'quantile', 'kmeans'}, default='quantile'*

Strategy used to define the widths of the bins.

- 'uniform': All bins in each feature have identical widths.
- 'quantile': All bins in each feature have the same number of points.
- 'kmeans': Values in each bin have the same nearest center of a 1D k-means cluster.

Motifs Fréquents et Règles d’association - Apriori

Quiz

1 - What is the main goal of the Apriori algorithm?

a) Clustering data

b) Finding frequent itemsets and association rules

c) Classifying datasets

d) Reducing dimensionality

Motifs Fréquents et Règles d’association - Apriori

Quiz

2 - What happens if you increase the minimum confidence threshold in Apriori?

a) More rules are generated

b) No effect on the number of rules

c) Fewer rules are generated

d) The dataset is modified

Motifs Fréquents et Règles d’association - Apriori

Quiz

3 - Which library is used for Apriori implementation in the notebook?

a) mlxtend

b) apriori_python

c) scikit-learn

d) pandas

Motifs Fréquents et Règles d’association - Apriori

Quiz

4- Which data structure is used to store transacts in apriori_python?

a) Dictionaries

b) List of Lists

c) Sets

d) DataFrames

Motifs Fréquents et Règles d'association - Apriori

Quiz

5- What is the function of the apriori() method in the notebook?

a) Preprocess the dataset

b) Generate frequent itemsets and association rules

c) Split the dataset into training and test sets

d) Visualize transaction patterns

Motifs Fréquents et Règles d’association - Apriori

Quiz

6- Which of the following is a valid reason for discretizing continuous attributes before applying Apriori?

a) Apriori works only with categorical data

b) Discretization improves algorithm efficiency

c) It allows grouping similar values into meaningful intervals

d) All of the above

Motifs Fréquents et Règles d'association - Apriori

Quiz

7- Which of the following statements is TRUE about pd.cut() in the context of discretization?

a) It assigns each value a unique integer identifier

b) It normalizes numerical values before discretizing

c) It replaces missing values with the bin's median

d) It creates bins of equal width across the range of values

Motifs Fréquents et Règles d’association - Apriori

Quiz

8- Which of the following is NOT a discretization strategy?

a) Equal-width binning

b) One-hot encoding

c) Equal-frequency binning

d) K-means binning

Motifs Fréquents et Règles d'association - Apriori

Quiz

9- Which of the following instruction can be used to retrieve the confidence of the last generated association rule?

a) **rules[-1][0]**

b) **rules[-1][-2]**

c) **rules[2][-1]**

d) **rules[-1][2]**

```
[[{'P3'}, {'P2'}, 0.6],
 [{'P2'}, {'P1'}, 0.6666666666666666],
 [{'P2'}, {'P1', 'P3'}, 0.6666666666666666],
 [{'P2', 'P3'}, {'P1'}, 0.6666666666666666],
 [{'P3'}, {'P1'}, 0.8],
 [{'P2'}, {'P3'}, 1.0],
 [{'P1'}, {'P3'}, 1.0],
 [{'P1', 'P2'}, {'P3'}, 1.0]]
```

Motifs Fréquents et Règles d’association - Apriori

Quiz

10- What happens if a value in the dataset falls outside the specified bins in pd.cut()?

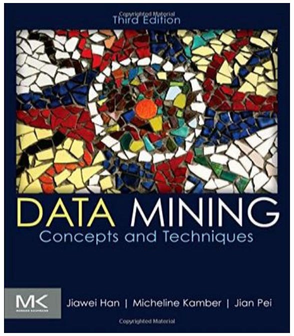
a) It is assigned to the closest bin

b) It is assigned to a new bin created automatically

c) It is assigned NaN

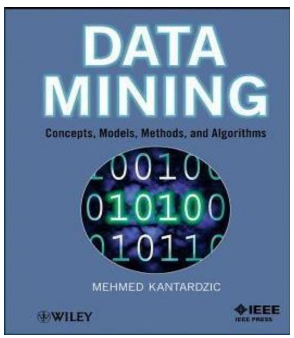
d) انا شا دخلني

Ressources



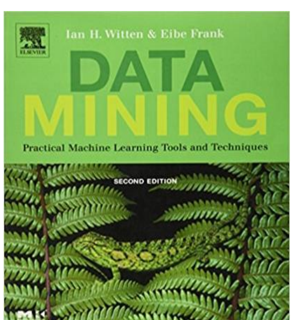
Data Mining : concepts and techniques, 3rd Edition

- ✓ Auteur : Jiawei Han, Micheline Kamber, Jian Pei
- ✓ Éditeur : Morgan Kaufmann Publishers
- ✓ Edition : Juin 2011 - 744 pages - ISBN 9780123814807



Data Mining : concepts, models, methods, and algorithms

- ✓ Auteur : Mehmed Kantardzi
- ✓ Éditeur : John Wiley & Sons
- ✓ Edition : Aout 2011 – 552 pages - ISBN : 9781118029121



Data Mining: Practical Machine Learning Tools and Techniques

- ✓ Auteur : Ian H. Witten & Eibe Frank
- ✓ Éditeur : Morgan Kaufmann Publishers
- ✓ Edition : Juin 2005 - 664 pages - ISBN : 0-12-088407-0

Motifs Fréquents et Règles d’association - Apriori

Larger CSV Dataset - Market Basket Analysis

```python
# Access the first two rows
rows = df[0:2]
rows
```

	0	1	2	3	4	5	6	7	8	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tom ju
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	N

```python
# Access the first and second column
df[[0, 1]]
```

	0	1
0	shrimp	almonds
1	burgers	meatballs
2	chutney	NaN
3	turkey	avocado
4	mineral water	milk
...	...	...
7496	butter	light mayo
7497	burgers	frozen vegetables
7498	chicken	NaN
7499	escalope	green tea
7500	eggs	frozen smoothie