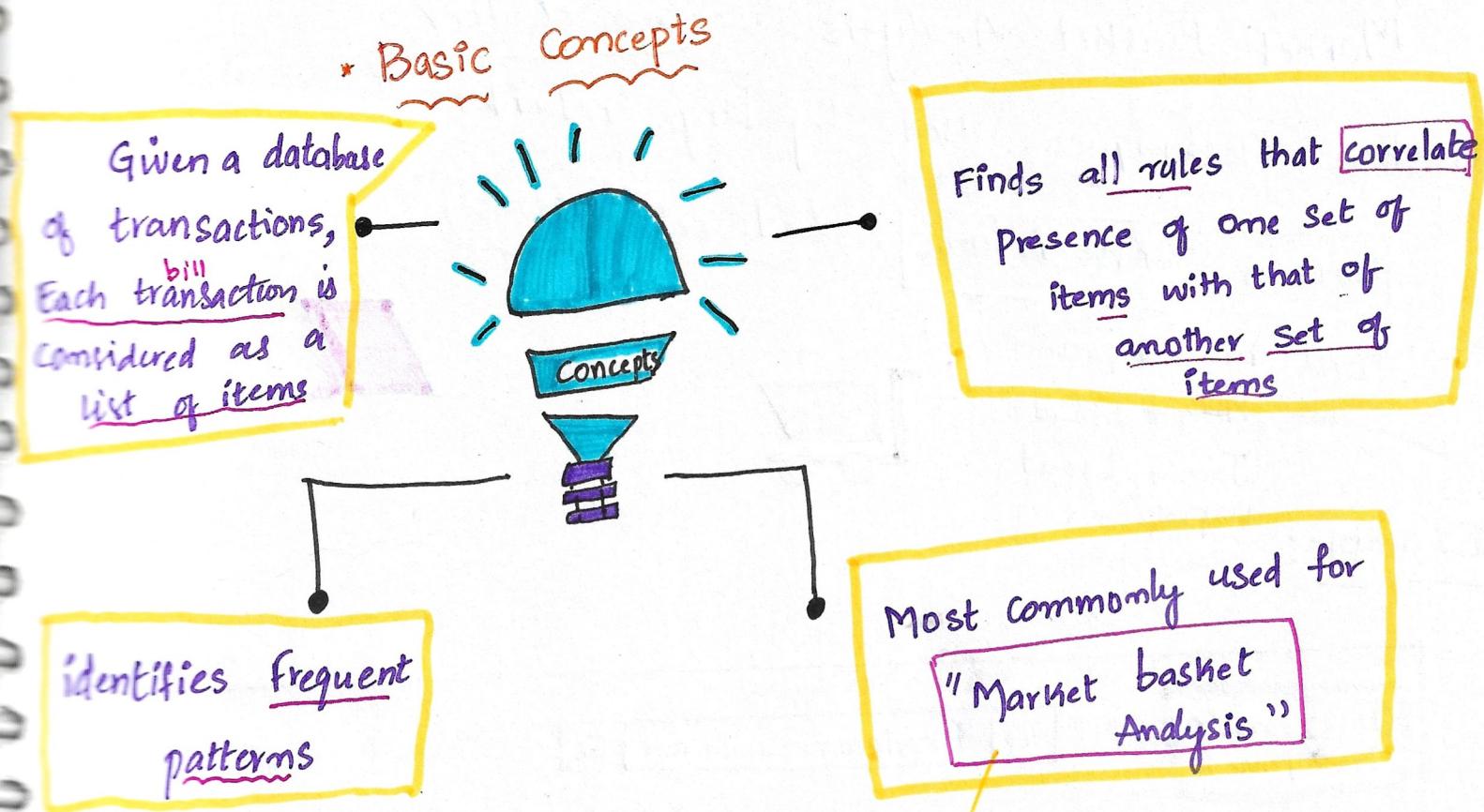


Association Rule

→ connection, relationship, correlation of things.



* ^{we} use association rules ?

- Super Markets
- E-commerce website
 - Amazon
 - flipkart
 - big basket

MBA → "Market Basket Analysis"

Set of Items

"Market Basket Analysis is one of the key Techniques used by large retailers to

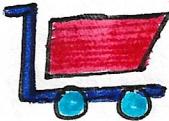
UnCover associations

Ex:- brush + paste

Milk + bread

Jam + bread

Laptop + bag

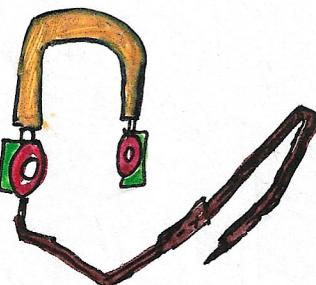
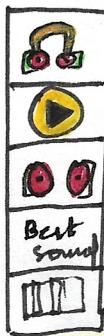


Example:

amazon.in
→ Prime

All → headphones with mic

< back to results

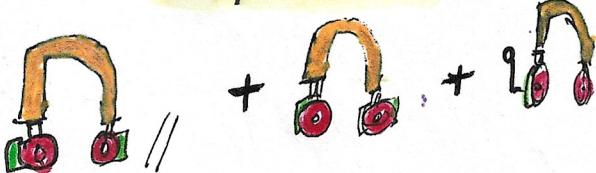


boat XXX900



₹ 649/-

Buy it with



Total price
₹ 1,897.00

Add all three care

This part
is called
"association
rules"

Date:- 13/06/22.

Association analysis :-

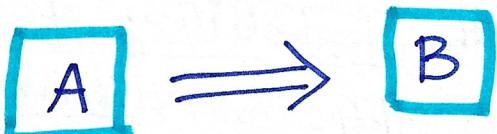
it's discovers the probability of occurrence of items in a collection. Helps in discovering some interesting relationships in large datasets.

A dataset contains data objects and each data object contains a set of attributes. An attribute is also called as "dimension" or "feature" or "variable" which represents the characteristics of a data object.

Ex:- Height, qualification, colour etc..

* Association Rule Mining :-
it finds the interesting associations and relation ships among large sets of data items. This rule shows how a item set occurs in a transaction.

* Association Rule :- Mining



IF

THEN

* Measure Association

1. Support
2. Confidence
3. Lift

* **Support** :- Number of transactions consists of A, B

total number of transactions

$$\text{Support} \Rightarrow \frac{\text{freq}[AB]}{N}$$

* **Confidence** :- Number of transactions consists of A, B

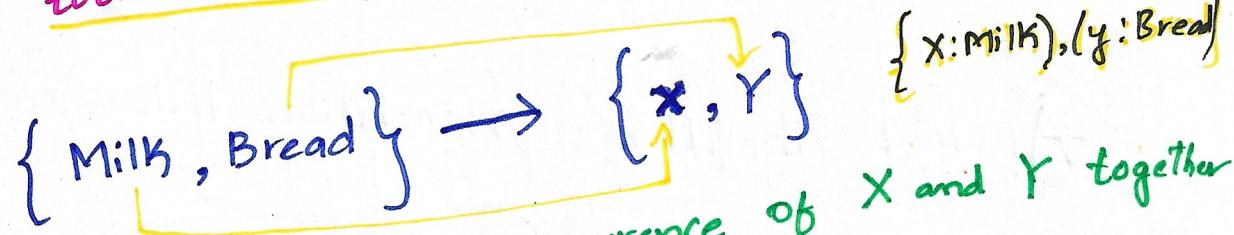
Number of transaction of only A

$$\star \text{ Confidence} \Rightarrow \frac{\text{freq}(A, B)}{\text{freq}(A)}$$

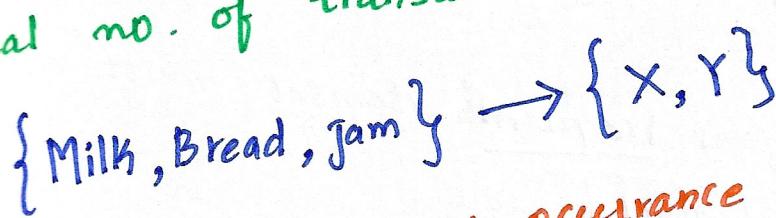
$$\star \text{ Lift} = \frac{\text{Support}}{\text{Support}(A) \times \text{Support}(B)}$$

Support

- it is a Measure of how frequently a Set of items occur in total number of transactions.



- Therefore the frequency of occurrence of X and Y together in total no. of transactions is support.



- Here with the frequency of occurrence of {Bread, Jam} in whole transaction is "support".

$$\text{Support}(s) = \frac{|(X \cup Y)|}{N}$$

Confidence

- It is a Measure of how often items in "Y" appears in transactions that contain "X"
- $\{ \text{Milk, Bread, Jam} \} \rightarrow \{ X, Y \}$ $(X: \text{Milk}, Y: \text{Bread, Jam})$

- Therefore The frequency of occurrence of X and Y in all transactions where "X" exists.

$$\text{Confidence } (c) = \frac{\sigma(X \cup Y)}{\sigma X}$$

* Apriori Algorithm: Association Rule Generation.

1 Uses frequency itemset to generate association rule

2 A subset of a frequent itemset must also be a frequent itemset

3 Frequent itemset is a set of items whose support value is greater than threshold value

Association Rule mining:

Given a set of transactions T , the goal of association rule mining is to find all rules

having

- Support \geq min Support threshold
- Confidence \geq min Confidence threshold

Example :-

Market Basket Data

(Bill) Transaction ID	items
1	{Milk, Bread, Rice, Book}
2	{Bread, Jam, Books, Pen}
3	{Jam, Milk, Bread, Rice, Eggs}
4	{Rice, Eggs, Pen, Book}
5	{Eggs, Pen, Milk, Bread, Jam}
6	{Eggs, Rice, Bread, Jam}

* Unique items :- {Milk, Bread, Jam, Rice, Eggs, Book, Pen}

Converting into "Dummies"

Transaction ID	Items						
	Milk	Bread	Jam	Rice	Eggs	Book	Pen
1	1	1	0	1	0	1	0
2	0	1	1	0	0	1	1
3	1	1	1	1	1	0	0
4	0	0	0	1	1	1	1
5	1	1	1	0	1	0	1
6	0	1	1	1	1	0	0

→ Frequent itemset :-

- * An itemset contains "k" items.
Then it is known as k itemset
- Set of products bought frequently
- * if it has only "milk" items. Then it is called "one itemset"
- * {"Milk" and "Bread"} = 2 (two) itemset.
- * {Milk, Bread, Jam} = 3 (three) itemset

Ex:- Frequent itemsets.

- Two itemsets : $\{\text{Milk, Bread}\}$, $\{\text{Bread, Jam}\}$, $\{\text{Rice, Egg}\}$
 $\{\text{Books, Pen}\}$
- Three itemsets : $\{\text{Milk, pen, Book}\}$, $\{\text{Rice, Bread, Eggs}\}$
 $\{\text{Book, Eggs, Pen}\}$
- Four itemsets : $\{\text{Milk, Bread, Rice, Eggs}\}$ Etc.....

~~2/2/13~~
13/06/22

Dt:- 14/06/22

Example :- 01

Suppose,

$$\begin{aligned} \text{min Sup} &= 0.3 \\ (\text{min. support}) & \\ \text{min Conf} &= 0.6 \\ (\text{min. confidence}) & \end{aligned}$$

Taken Randomly
We can change by Hyper parameter Tuning

TID	items
1	$\{\text{Milk, Bread, } \boxed{\text{Rice}}, \text{ Book}\}$
2	$\{\text{Bread, Jam, Book, Pen}\}$
3	$\{\text{Jam, Milk, Bread, } \boxed{\text{Rice, Eggs}}\}$
4	$\boxed{\text{Rice, Eggs}}, \text{ Pen, Book}$
5	$\{\text{Eggs, Pen, Milk, Bread, Jam}\}$
6	$\boxed{\text{Eggs, Rice}}, \text{ Bread, Jam}$

Consider.

$$\{\text{Rice, Eggs}\} \rightarrow \{x, y\}$$

Support (S) =

$$\frac{|(x \cup y)|}{N}$$

$$\text{Support (S)} = \frac{1+1+1}{6} = \frac{3}{6} = 0.5$$

↑
Total no. of

it checks How many no. of Transactions consist of "Rice," "Eggs"

$$\text{Confidence } (c) = \frac{\sigma(X \cup Y)}{\sigma X} \rightarrow \frac{\text{No. of items present in } X, Y}{\text{No. of transaction in } X}$$

$\text{confidence } (c) = \frac{3}{4} = 0.75$

↑ Eggs + Rice
↓ Rice

* Association Rule Mining : If we got 0.2, we take this combination.

Here, Support (s) : $0.5 \geq \text{min Sup } (0.3)$

Confidence (c) : $0.75 \geq \text{min Conf } (0.6)$

Therefore, we can mine

{Rice, Eggs} as a rule.

Suppose, we have three (3) items

Example : 2

Suppose,

$$\text{min Sup} = 0.3$$

$$\text{min Conf} = 0.6$$

Consider,

$$\{Milk, Bread, Jam\} \rightarrow [X, Y]$$

TID	Items
1	{Milk, Bread, Rice, book}
2	{Bread, Jam, Book, Pen}
3	{Jam, Milk, Bread, Rice, Eggs}
4	{Rice, Eggs, Pen, Book}
5	{Eggs, pen, Milk, Bread, Jam}
6	{Eggs, Rice, Bread, Jam}

$$\text{Support}(S) = \frac{\sigma(X \cup Y)}{N}$$

$$\text{Support}(S) = \frac{1+1}{6} = \frac{2}{6} = 0.333$$

and,

$$\text{Confidence}(C) = \frac{\sigma(X \cup Y)}{\sigma X}$$

$$\text{Confidence}(C) = \frac{2}{3} = 0.667$$

If a person bought $\{ \text{milk}, \text{bread} \}$ → what is the probability he's going to buy $\{ \text{jam} \}$

* Association Rule Mining:

$$\text{Support}(S) : 0.333 \geq \text{minSup} (0.3)$$

$$\text{Confidence}(C) : 0.667 \geq \text{minConf} (0.6)$$

Therefore, we can mine

$\{ \text{Milk}, \text{Bread}, \text{jam} \}$ as a rule.

Code :-

Here, we take the same Example in Python coding

```
# Data = [[ "milk", "bread", "rice", "book"],  
[ "bread", "Jam", "book", "Pen"],  
[ "Jam", "Milk", "bread", "rice", "eggs"],  
[ "rice", "Eggs", "pen", "books"],  
[ "Eggs", "pen", "milk", "bread", "Jam"],  
[ "Eggs", "rice", "bread", "Jam"]]
```

install : pip install mlxtend

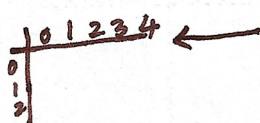
```
from mlxtend.preprocessing import TransactionEncoder  
↳ it converts the  
data into array
```

```
# te = TransactionEncoder()
```

```
# te_array = te.fit_transform(Data)
```

```
import pandas as pd
```

```
# df = pd.DataFrame(te_array, columns = te.columns_)
```



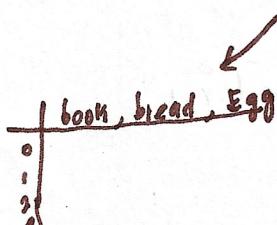
- If we don't write this one.
it gives (0, 1, 2)

- it takes in alphabetical order

book, bread, eggs, Jam, milk, pen

```
# df
```

dataframe



Out:

	book	bread	eggs	Jam	milk	Pen	rice
0	True	True	False	False	True	False	True
1	True	True	False	True	False	True	False
2	False	True	True	True	True	False	True
3	True	False	True	False	False	True	True
4	False	True	True	True	True	True	False
5	False	True	True	True	False	False	True

Date: 20/6/22
11:00 AM

14/06/22

from mlxtend.frequent_patterns import apriori

itemset = apriori (df, min_support = 0.3 , use_colnames = True)

itemset

Out:	Support	itemsets
	0.500000	(book)
	0.833333	(bread)
	0.666667	eggs
	0.333333	(bread, book)
	0.333333	(book, rice)
	0.500000	(bread, Jam, Eggs)..... 33 results

We can take anything (Randomly)
0.5, 0.6, ...
Theokovitch

from mlxtend.frequent_patterns import association_rules

res = association_rules(itemset, metric = "confidence", min_threshold = 0.6)

res

Out:	if ↓ antecedents	Then ↓ Consequents	antecedent-support	consequent-support	support	confidence	Lift
0	book	bread	0.500000	0.833333	0.333333	0.666667	0.800000
1	book	pen	0.500000	0.500000	0.333333	0.666667	1.333333
2	pen	book	0.500000	0.500000	0.333333	0.666667	1.333333
....
69	(bread_rice)	(jam, eggs)	0.500000	0.500000	0.333333	0.666667	1.333333
70	(bread_eggs)	(rice, jam)	0.500000	0.333333	0.333333	0.666667	2.000000
71	(rice_jam)	(bread, Eggs)	0.333333	0.500000	0.333333	1.000000	2.000000
72	(rice_eggs)	(bread, Jam)	0.5000000	0.500000	0.333333	0.666667	1.000000

→ To Reduce options (or
To gain accuracy, we
change The min-Support = 0.6
min-threshold = 0.6)

→ it Gives The less item by
Sorting the with probability.

Leverage	Conviction
-0.08333	0.500000
0.08333	1.500000
0.08333	1.500000
....
0.08333	1.500000
0.16667	2.000000
0.16667	if 1.000000
0.00000	

again

from mlxtend.frequent_patterns import apriori

itemSet = apriori (df, min-Support = 0.6,
use_colnames = True)

itemset

[out]:

	Support	itemsets
0	0.833333	bread
1	0.666667	eggs
2	0.666667	Jam
3	0.666667	rice
4	0.666667	(bread, Jam)

from mlxtend.frequent_patterns import association_rules

res = association_rules (itemset, metric = "confidence",
min_threshold = 0.6)

res if then

	antecedes	Consequents	antecedes Support	consequents Support	Support Confiden- ce	Lift	Leverag- e	Convic- tion
0	bread	Jam	0.833333	0.666667	0.666667	0.8	1.2	0.111111 1.666667
1	Jam	bread	0.666667	0.833333	0.666667	1.0	1.2	0.111111 inf

$$\frac{\text{bread, Jam}}{\text{Total}} = \frac{4}{6} = \frac{2}{3} = 0.666667$$

We Only Consider only Few column for Gaining Result

result = res [["antecedents", "consequents", "support",
"confidence", "lift"]]

result

out:	antecedents	Consequents	Support	Confidence	Lift
0	bread	Jam	0.666667	0.8	1.2
1	Jam	bread	0.666667	1.0	1.2

result [result ["confidence"] >= 1]

out:	antecedents	Consequents	Support	Confidence	Lift
	Jam	bread	0.666667	1.0	1.2

