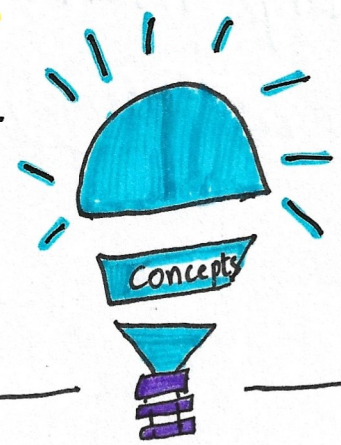
Dt:- 18/5/22
10:30 AM

Association Rule

↳ connection, relationship, correlation of things.

* Basic Concepts



Given a database of transactions, Each transaction (bill) is considered as a list of items

Finds all rules that correlate Presence of one set of items with that of another set of items

identifies frequent patterns

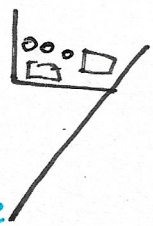
Most commonly used for "Market basket Analysis"

* Were we use association rules ?
  - Super Markets
  - E-commerce website
    - Amazon
    - flipkart
    - bigbasket

MBA → "Market Basket Analysis"

↓

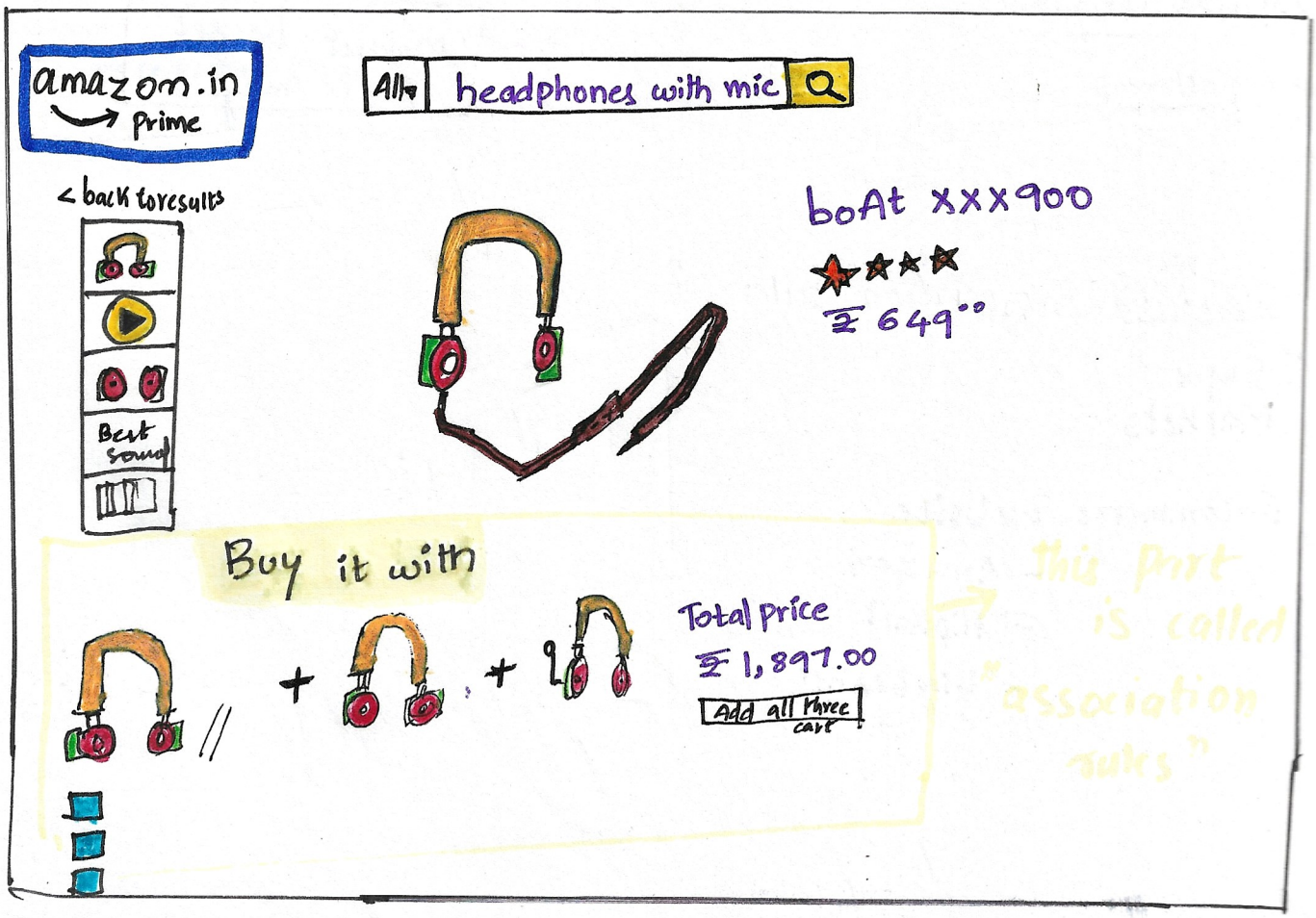
Set of items



" Market Basket Analysis is one of the Key Techniques used by large retailers to Uncover associations between items.

Ex:- brush + paste
Milk + bread
Jam + bread
Laptop + bag

Example:



Date:- 13/06/22.

Association analysis :-

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

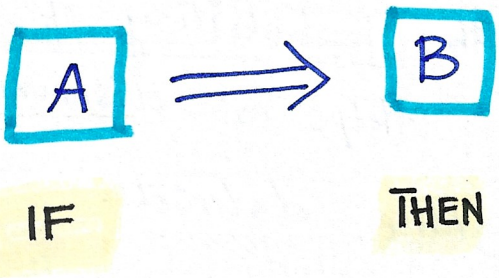
A dataset Contains data objects and Each data object Contains a set of attributes. an attributes is also called as "dimension" or "Feature" or "variable" which represents the characterstics of a data object.

Ex :- Height, Qualification, Colour etc...

Association Rule Mining :-

it finds the interstng associatons and relation-ships among large sets of data items. this rule shows how frequently a itemset occurs in a transaction.

* Association Rule Mining :-



* Measure * Association

1. Support
2. Confidence
3. Lift

* **Support** :- $\frac{\text{Number of transactions Consists of } A, B}{\text{total number of transactions}}$

$$\text{Support} \Rightarrow \frac{\text{freq}[A,B]}{N}$$

* **Confidence** :- $\frac{\text{Number of transactions Consists of } A, B}{\text{Number of transaction of only } A}$

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

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

Support

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

$\{Milk, Bread\} \longrightarrow \{X, Y\}$ $\{X: Milk\}, \{Y: Bread\}$

- There fore the frequency of occurence of X and Y together in total no. of transactions is support.

$\{Milk, Bread, Jam\} \longrightarrow \{X, Y\}$ $\{X: Milk\}, \{Y: Bread, Jam\}$

- Here the Frequency of occurance of {Bread, Jam} with {Milk} in whole transaction is "support."

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

Confidence

- It is a Measure of how often items in "Y" appears in transactions that contain "X"

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

- Therefore The Frequency of occurance 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 > Threshold Value.

Greater Than

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, Book, 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 :-

* Set of Products bought frequently

→ An itemset contains "K" items, Than it is known as K itemset

* if it has only "milk" items, Than it is called "one itemset"
* {"Milk" and "Bread"} = 2 (two) itemset.
* {Milk, Bread, Jam} = 3 (three) itemset

Ex:- Frequent itemsets.

- Two itemsets : {Milk, Bread}, {Bread, Jam}, {Rice, Eggs}
  {Books, Pen}
- Three itemsets : {Milk, Pen, Book}, {Rice, Bread, Eggs}
  {Book, Eggs, Pen}
- Four itemsets : {Milk, Bread, Rice, Eggs} Etc.....

15/06/22

Dt:- 14/06/22

Example :- 01

Suppose,

min Sup (min. support) = 0.3 (Taken Randomly)

min conf (min. confidence) = 0.6

We can change by Hyper parameter Tunning

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}

Consider.

{Rice, Eggs} → {x, y}

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

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

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

↳ Total no. of

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

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

(3 → Eggs + Rice; 4 → Rice)

* **Association Rule Mining :-**

Here, Support (s) : 0.5 ≥ min Sup (0.3) ("if we got 0.2, we take this combination.")

Confidence (c) : 0.75 ≥ min Conf (0.6)

Therefore, We can Mine

{Rice, Eggs} as a rule.

Example : 2 Suppose, we Have Three (3) items

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}

Suppose,

min Sup = 0.3

min Conf = 0.6

Consider,

{Milk, Bread, Jam} → {X, Y}

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

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

and,

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

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

(2 → milk + bread, Jam; 3 → {milk + Bread})

If a person bought {milk + Bread} → what is the probability he's going to buy {"Jam"}

* Association Rule mining :

Support (s) : $0.333 \geq$ minsup (0.3)

Confidence (c) : $0.667 \geq$ min conf (0.6)

There fore, we can Mine

{Milk, Bread, 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", "book"],
          ["Eggs", "pen", "milk", "bread", "Jam"],
          ["Eggs", "rice", "bread", "Jam"]]
```

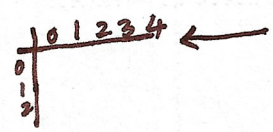
install : pip install mlxtend

```
from mlxtend.preprocessing import TransactionEncoder
```

↳ it converts the data into array

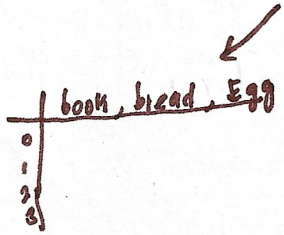
```
# te = TransactionEncoder()
# te_array = te.fit_transform(Data)

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

- If we don't write this one, it gives (0, 1, 2, 3, 4)



- it Takes in alphabetical order

book, bread, eggs, Jam, milk, pen



Out : dataframe

	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

06/22

Date:- 20/6/22
11:00Am

```
from mlxtend.frequent_patterns import apriori

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

# itemset
```

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

Out:

Support	itemsets
0.500000	(book)
0.83333	(bread)
0.66667	eggs
0.333333	(bread, book)
0.333333	(book, rice)
0.500000	(bread, Jam, Eggs) ..... 33 results

```
from mlxtend.frequent_patterns import association_rules

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

# res
```

Out:

	antecedents (if)	consequents (Then)	antecedent_support	consequent_support	support	confidence	Lift	Leverage	Conviction
0	book	bread	0.500000	0.833333	0.33333	0.666667	0.8000000	-0.08333	0.500000
1	book	pen	0.500000	0.500000	0.33333	0.66667	1.333333	0.08333	1.500000
2	pen	book	0.500000	0.500000	0.33333	0.66667	1.333333	0.08333	1.500000
	....	....	.....	.....	.....	.....	.....	.....	.....
69	(bread_rice)	(jam, eggs)	0.500000	0.500000	0.33333	0.666667	1.333333	0.08333	1.5000000
70	(bread_eggs)	(rice, jam)	0.500000	0.33333	0.33333	0.666667	2.000000	0.166667	2.000000
71	(rice_jam)	(bread, Eggs)	0.33333	0.500000	0.33333	1.000000	2.000000	0.000000	if
72	(rice_eggs)	(bread, Jam)	0.5000000	0.500000	0.33333	0.666667	1.000000		1.0000000

⇒ To Reduce options (or more probability) To gain accuracy, we change the

min_support = 0.6
min_threshold = 0.6

⇒ it Gives the less item by sorting the high probability.

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

out:

	antecedes	consequents	antecedes support	consequents support	support	confidence	lift	leverage	conviction
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 [[ "antecedes", "consequents", "support",
                 "confidence", "Lift" ]]
```

```
# result
```

Out:

	antecedents (If)	Consequents (then)	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]
```

Greater Than = 1

Out:

antecedents (If)	Consequents (then)	Support	Confidence	Lift ("Support" or "Lift" → we can Give our own options there)
Jam	bread	0.666667	1.0	1.2

highest the lift ⇓ higher the combination

items are selected by Given Support & Confidence

offers are designed by using This method in Super markets are any Grocery Stores

no. of times it repeated (or) not it applies Concept of no. of times repeation (Frequency)

min_support, min_confidence is not fixed.

The more no. of times it repeated it has more association

it changes item by item, which gives high "probability"

it Gives combinations of items

For large data of items apply For loop