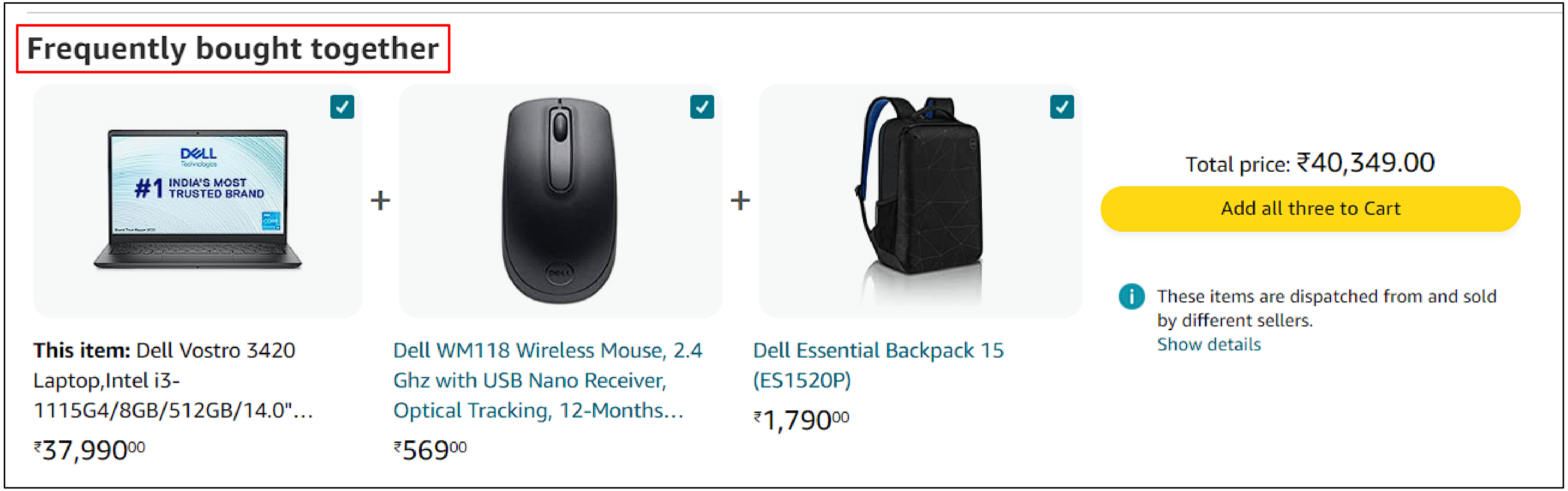
Association Rule Mining

UTKARSH GAIKWAD

Association Rule Mining Example

Amazon Items that are frequently bought together



Frequently bought together

☑ **This item:** Dell Vostro 3420 Laptop,Intel i3-1115G4/8GB/512GB/14.0"...
₹37,990^{00}

+

☑ Dell WM118 Wireless Mouse, 2.4 Ghz with USB Nano Receiver, Optical Tracking, 12-Months...
₹569^{00}

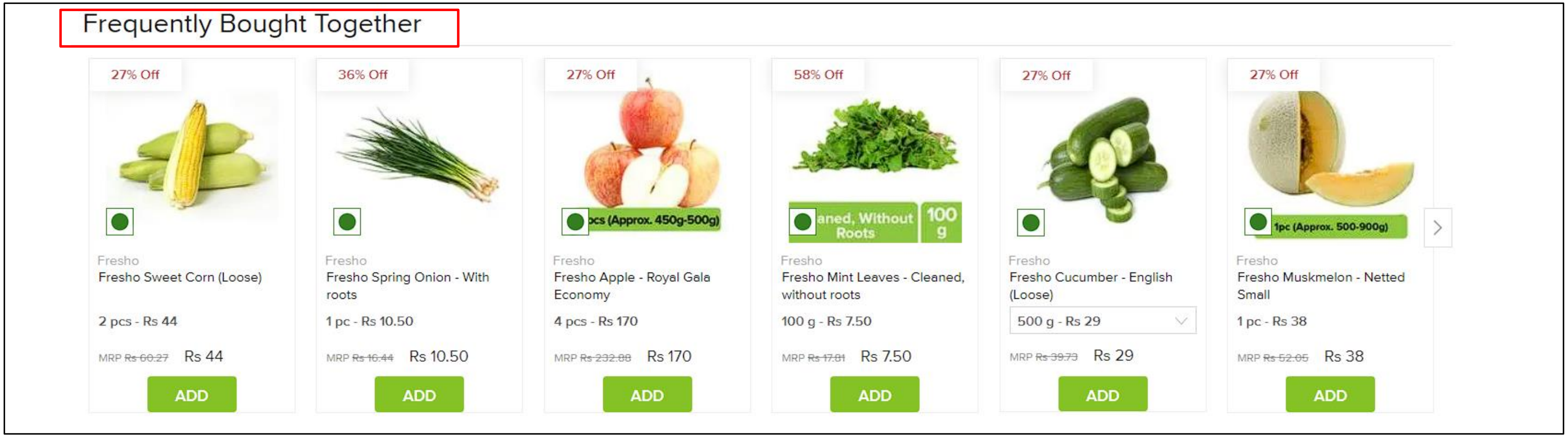
+

☑ Dell Essential Backpack 15 (ES1520P)
₹1,790^{00}

Total price: ₹40,349.00

These items are dispatched from and sold by different sellers.
Show details

Example of ARM in Big Basket website



Frequently Bought Together

27% Off
Fresho
Fresho Sweet Corn (Loose)
2 pcs - Rs 44
MRP ~~Rs 60.27~~ Rs 44

36% Off
Fresho
Fresho Spring Onion - With roots
1 pc - Rs 10.50
MRP ~~Rs 16.44~~ Rs 10.50

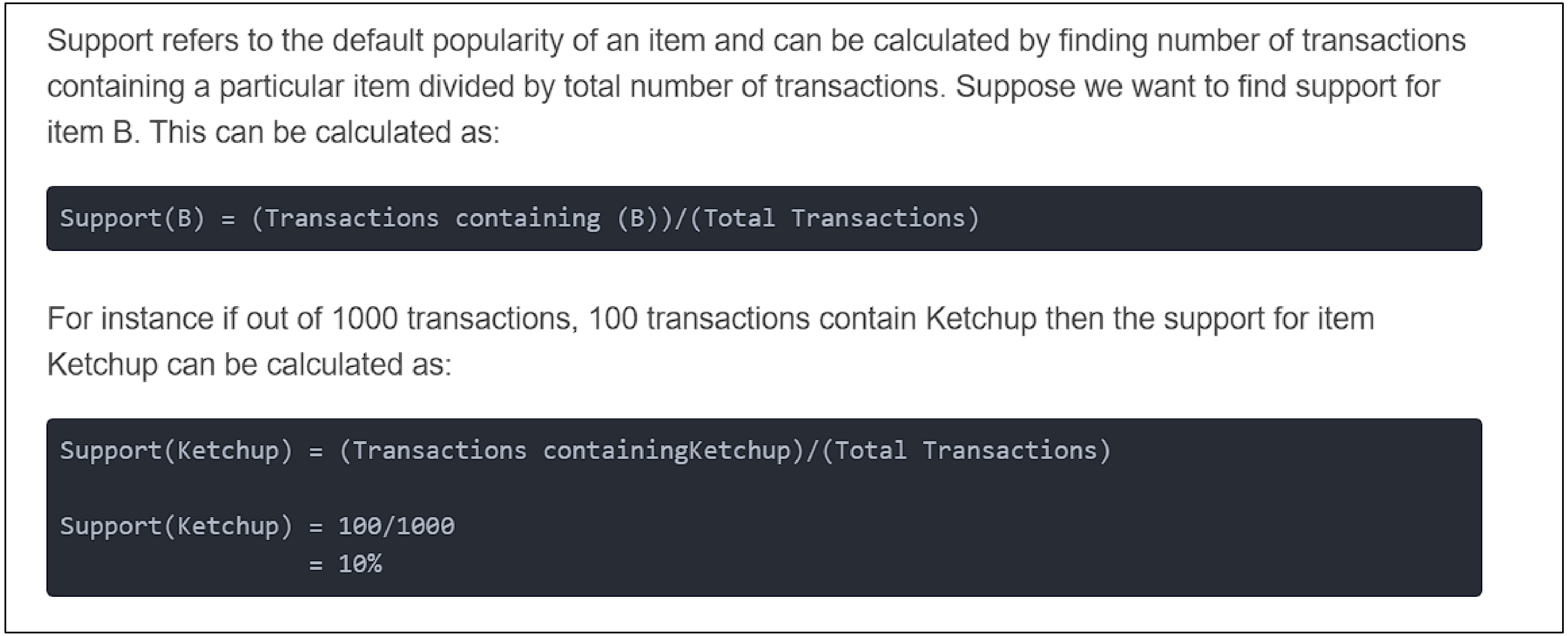
27% Off
Fresho
Fresho Apple - Royal Gala Economy
4 pcs - Rs 170
MRP ~~Rs 232.88~~ Rs 170

58% Off
Fresho
Fresho Mint Leaves - Cleaned, without roots
100 g - Rs 7.50
MRP ~~Rs 17.81~~ Rs 7.50

27% Off
Fresho
Fresho Cucumber - English (Loose)
500 g - Rs 29
MRP ~~Rs 39.73~~ Rs 29

27% Off
Fresho
Fresho Muskmelon - Netted Small
1 pc - Rs 38
MRP ~~Rs 52.05~~ Rs 38

Support



Support refers to the default popularity of an item and can be calculated by finding number of transactions containing a particular item divided by total number of transactions. Suppose we want to find support for item B. This can be calculated as:

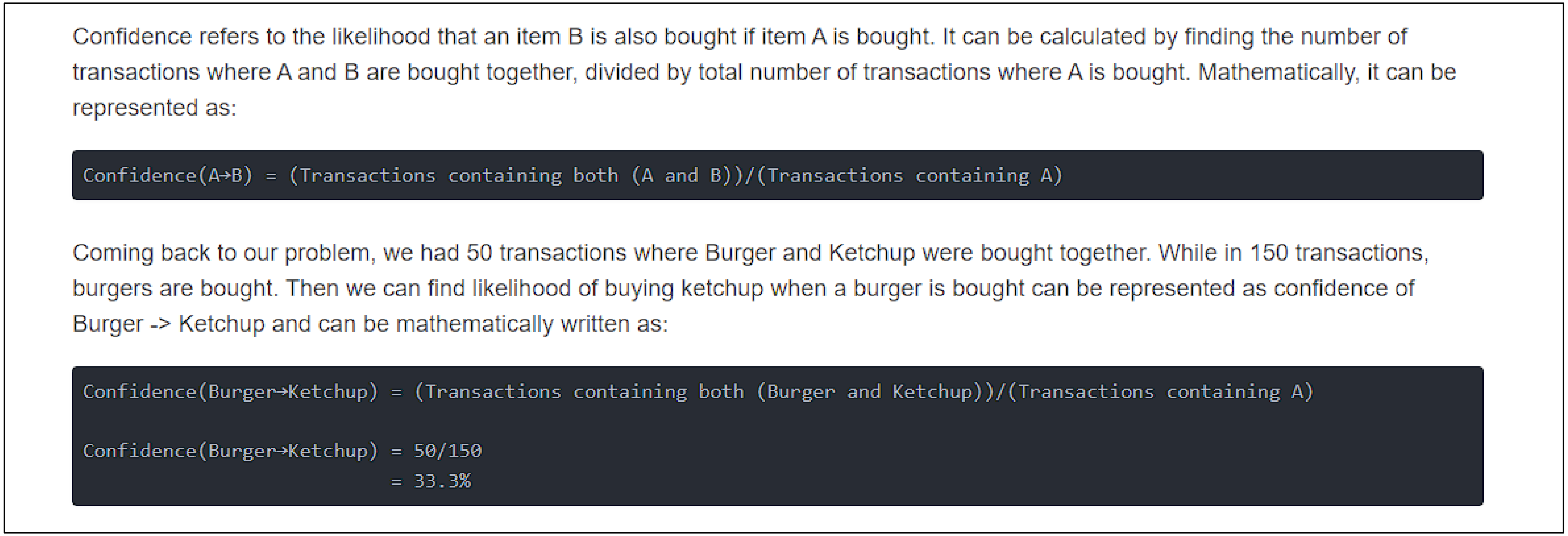
```
Support(B) = (Transactions containing (B))/(Total Transactions)
```

For instance if out of 1000 transactions, 100 transactions contain Ketchup then the support for item Ketchup can be calculated as:

```
Support(Ketchup) = (Transactions containingKetchup)/(Total Transactions)

Support(Ketchup) = 100/1000
                 = 10%
```

Confidence



Confidence refers to the likelihood that an item B is also bought if item A is bought. It can be calculated by finding the number of transactions where A and B are bought together, divided by total number of transactions where A is bought. Mathematically, it can be represented as:

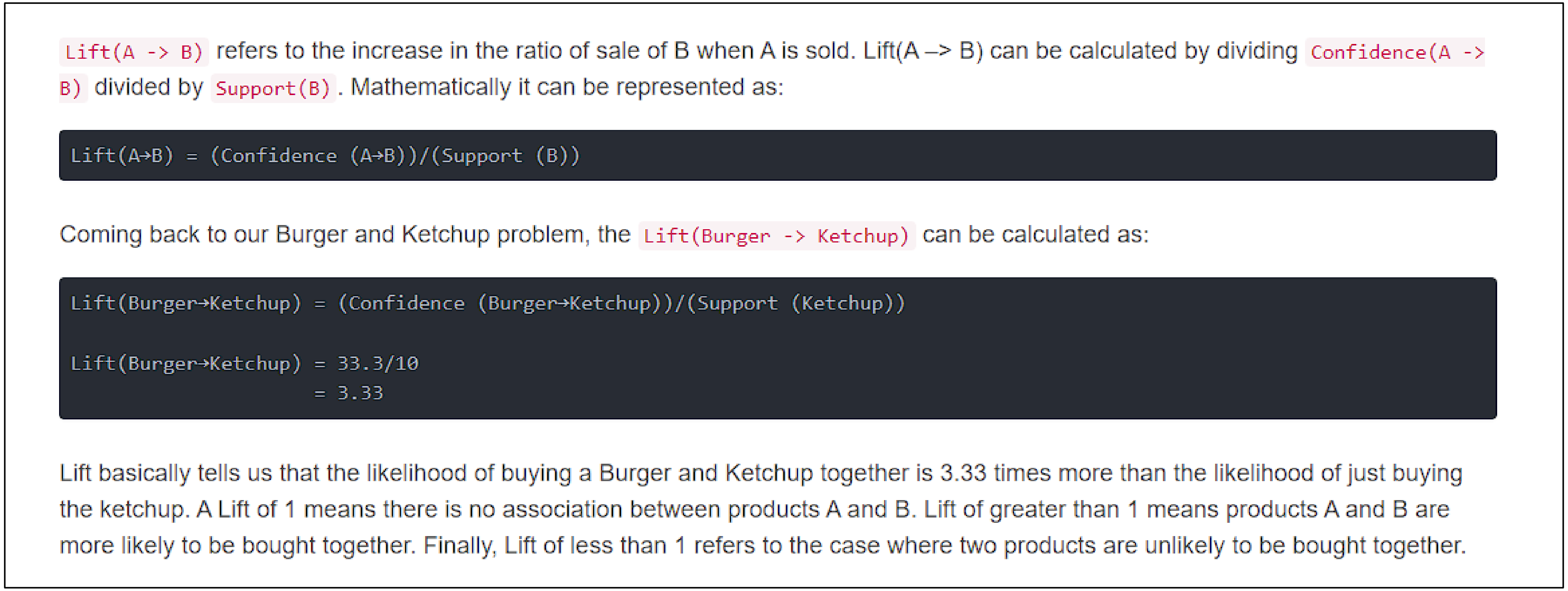
```
Confidence(A→B) = (Transactions containing both (A and B))/(Transactions containing A)
```

Coming back to our problem, we had 50 transactions where Burger and Ketchup were bought together. While in 150 transactions, burgers are bought. Then we can find likelihood of buying ketchup when a burger is bought can be represented as confidence of Burger -> Ketchup and can be mathematically written as:

```
Confidence(Burger→Ketchup) = (Transactions containing both (Burger and Ketchup))/(Transactions containing A)

Confidence(Burger→Ketchup) = 50/150
                           = 33.3%
```

Lift



`Lift(A -> B)` refers to the increase in the ratio of sale of B when A is sold. Lift(A –> B) can be calculated by dividing `Confidence(A -> B)` divided by `Support(B)`. Mathematically it can be represented as:

```
Lift(A→B) = (Confidence (A→B))/(Support (B))
```

Coming back to our Burger and Ketchup problem, the `Lift(Burger -> Ketchup)` can be calculated as:

```
Lift(Burger→Ketchup) = (Confidence (Burger→Ketchup))/(Support (Ketchup))

Lift(Burger→Ketchup) = 33.3/10
                     = 3.33
```

Lift basically tells us that the likelihood of buying a Burger and Ketchup together is 3.33 times more than the likelihood of just buying the ketchup. A Lift of 1 means there is no association between products A and B. Lift of greater than 1 means products A and B are more likely to be bought together. Finally, Lift of less than 1 refers to the case where two products are unlikely to be bought together.

Steps Involved in Apriori Algorithm

For large sets of data, there can be hundreds of items in hundreds of thousands transactions. The Apriori algorithm tries to extract rules for each possible combination of items. For instance, Lift can be calculated for item 1 and item 2, item 1 and item 3, item 1 and item 4 and then item 2 and item 3, item 2 and item 4 and then combinations of items e.g. item 1, item 2 and item 3; similarly item 1, item2, and item 4, and so on.

As you can see from the above example, this process can be extremely slow due to the number of combinations. To speed up the process, we need to perform the following steps:

1. Set a minimum value for support and confidence. This means that we are only interested in finding rules for the items that have certain default existence (e.g. support) and have a minimum value for co-occurrence with other items (e.g. confidence).

2. Extract all the subsets having higher value of support than minimum threshold.

3. Select all the rules from the subsets with confidence value higher than minimum threshold.

4. Order the rules by descending order of Lift.

Thank you

UTKARSH GAIKWAD