**Association Rule Mining & Apriori Algorithm**

Have you ever wondered how you get recommendations about various products, foods, ads, YouTube videos, etc.? The recommendations that you get is on the basis of the data that many sources collect your searches. On the basis of these data they apply some Machine Learning algorithm to provide the best recommendations. These all happens with help of Apriori Algorithm and Association Rule. So, let’s see how all this happens.

Organizations such as Flipkart, Amazon began mines data related to frequently bought items. So market basket analysis is one of the key techniques used by large retailers to uncover associations between items now. Example:- A customer who purchase bread have 60% likelihood to also purchase jam; customers who purchase laptops are more likely to purchase laptop bags as well.

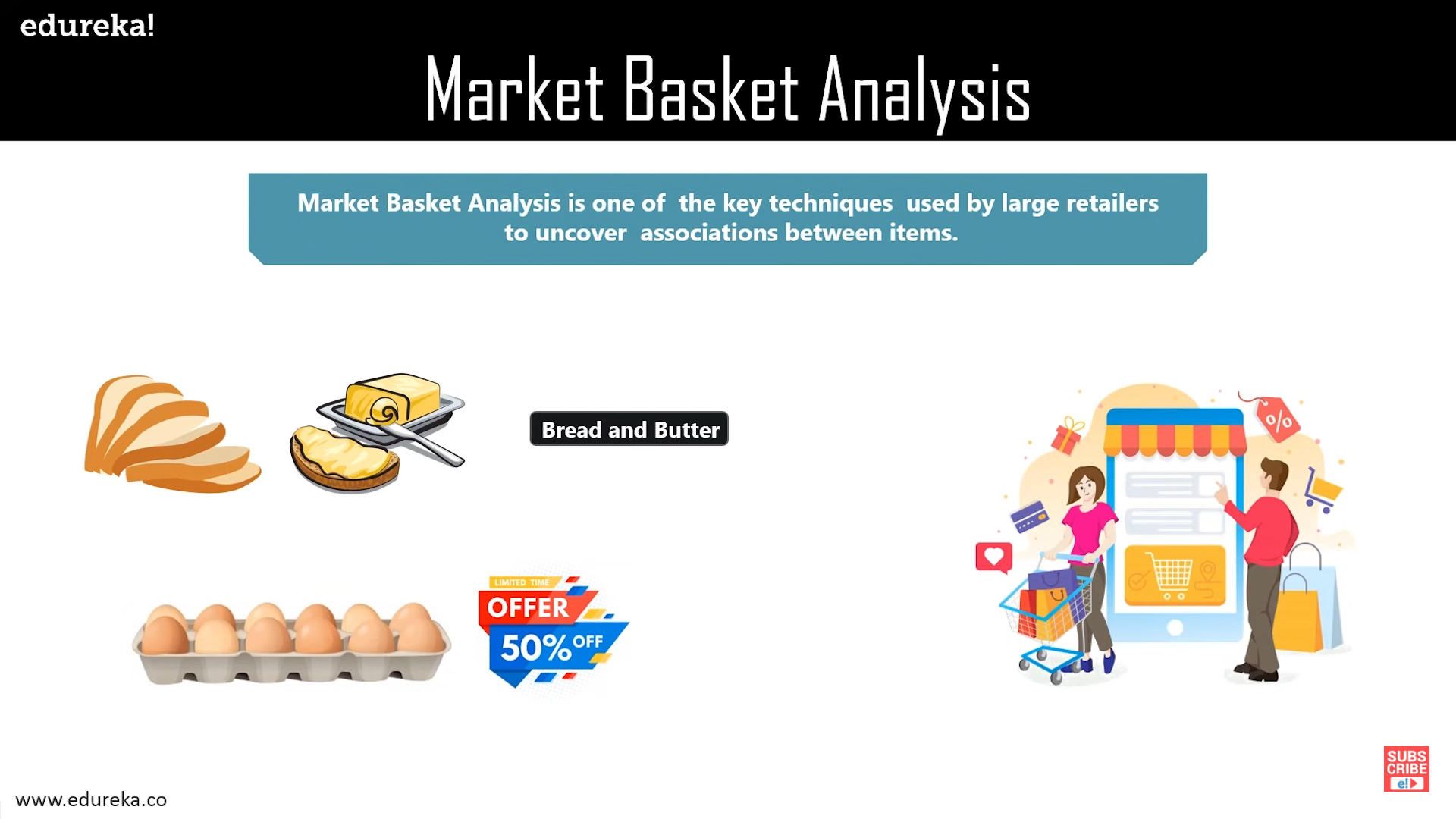
 

Fig.1 Bread and Jam Fig.2 Bread and Butter

These associations between different items and products that can be sold together which gives assisting in the right product placement. This helps marketing team target customers, like people who buy bread also tend to buy butter and on the basis of this marketing team can provide an offer to the customer to buy a third item suppose eggs. This all about market purchase analysis.

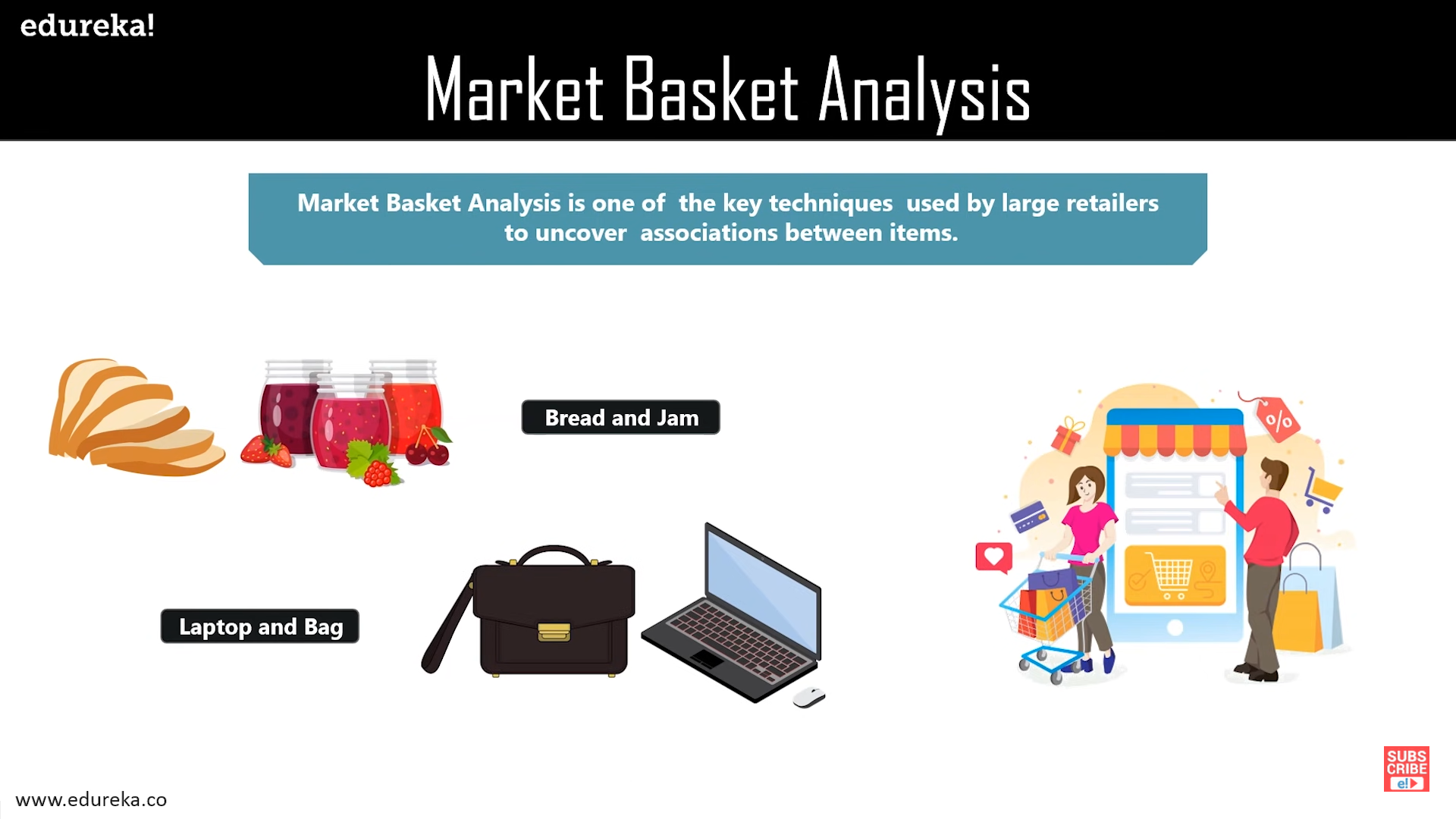


Fig.3 Market Purchase Analysis

Association rule can be thought of as an if-then relationship. Suppose if an item is being bought by the customer then the chances of item B picked by the customer is found out.

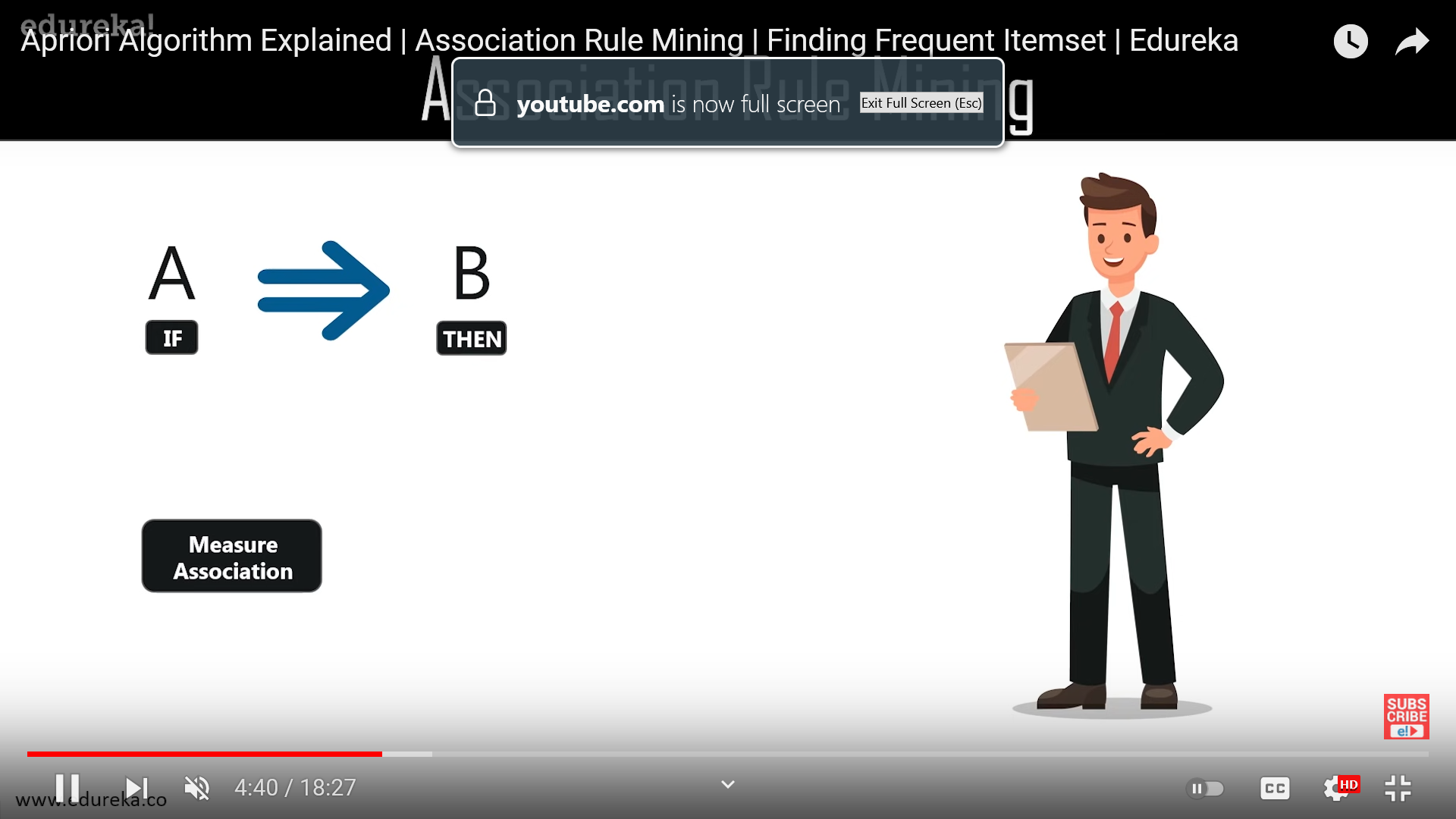


Fig.4 Co-occurrence pattern of A & B

It is a co-occurrence pattern that comes to play. Here are two elements first is “If” and second is “Then”. If is also known as antecedent this is an item or a group of items that are typically found in the item set and then is called the consequent. Consequent comes up along as an item with antecedent group or group of antecedent approaches.

The previous example was for only few items. Imagine if we have thousands of data or items and go to a data scientist we can make huge profits with proper insights of the data using Association Rule Mining.

The type of relationship in which we can find the relationship between two items (A=>B) is called single cardinality. Now, if the customer who bought A&B also wants to buy D cardinality usually increases, we can have a lot of combinations along these data. Now if we have around 10,000 or more than 10,000 data or items imagine how many rule we are going to create or deal with for each product. Thus we need association rule mining, so that we don’t end up with creating thousands of rules.

Maths and Logic behind Asosciation Rule Mining:-

There are three types of matrices which help to measure the association namely Support, Confidence, and Lift.

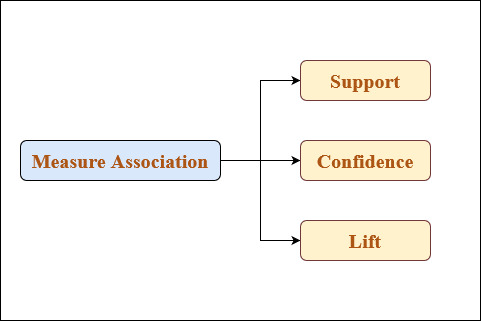


Fig.5 Types of measure Association

Support:-

Support is the frequency of item A or the combination of item A or B. It’s basically the frequency of the items which we have bought and what are the combination of the frequency of the items you have bought. So, with this you can filter out the items which we have bought less frequently.

Confidence:-

Confidence tells us how often the items A & B occur together given the number of items A & B occur together given the number of items A occur. This helps to solve problems when someone is buying A and B together and not C we can just rule out C at that point of time.

The advantages of using this two “support” and “confidence” is that it helps you to rule out a situation or problems which people just buy barely. So, according to this we can define a minimum support and confidence filter data out and create different rules.

**Note:-** Now suppose even after filtering we have 5000 thousand rules and for very item we create these five thousand rules. So, this is practically impossible. Thus, we need a third parameter “lift”.

Lift:-

Lift is basically the strength of any rule. In the denominator we have the independent support of A & B. This gives us the independent occurrence probability of A and B.

Now there is a lot of difference between random occurrence and association. If the denominator of the lift is more that means the occurrence of randomness is more rather than the occurrence of any association.

**Note:-** Lift is the final verdict where you decide to spend time on a particular rule.

A simple example on association rule mining:-

We have a set of items named A, B, C, D & E and a set of transactions T1, T2, T3, T4, T5. Let for each transactions consists of three items purchased.

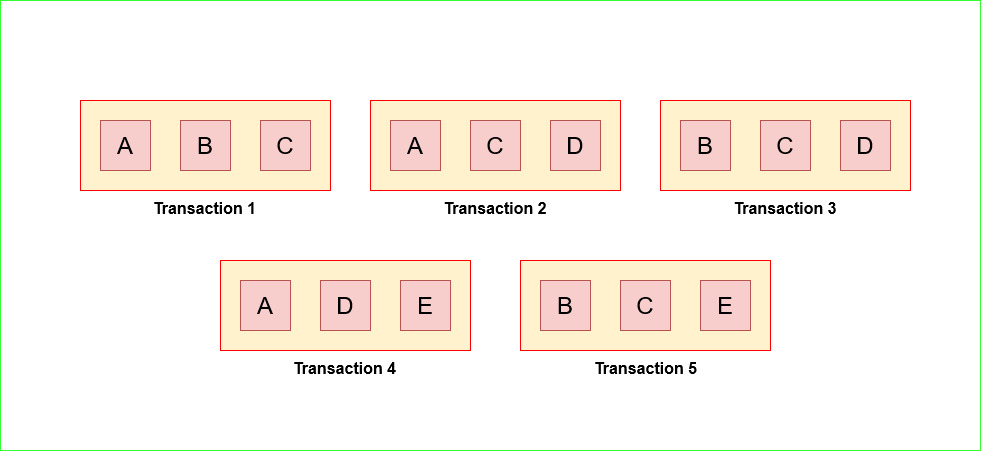


Fig.6 Five different transactions each consisting of three items

|  |  |  |  |
| --- | --- | --- | --- |
| **Transaction Id** | **Item Name** | **Item Name** | **Item Name** |
| T1 | A | B | B |
| T2 | A | C | D |
| T3 | B | C | D |
| T4 | A | D | E |
| T5 | B | C | E |

Table1. Tabular form of all the transactions along with the itesms for each case

Now based on these items and transactions we create some rules that specify or explain which item is brought with which one.

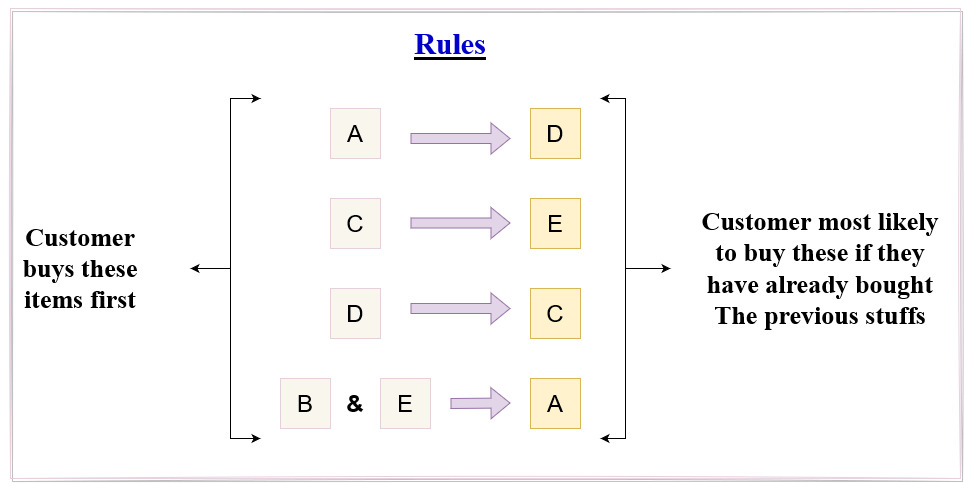


Fig.7 Some rules based on the transactions

Now, after calculating the support, confidence and lift of these rules we have a table and based on these values of Support, Confidence, and Lift you need to decide which set of rules you need to keep and which to exclude for providing the best recommendations to the customers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Rule** | **Support** | **Confidence** | **Lift** |
| A => D | 2/5 | 2/3 | 10/9 |
| C => E | 2/4 | 2/4 | 5/6 |
| F => C | 2/5 | 2/3 | 5/6 |
| B & E => A | 1/5 | 1/3 | 5/9 |

Table2. Tabular form of Support, Confidence and Lift of each Rule

Apriori Algorithm:-

Apriori Algorithm is an algorithm used for Association Rule Mining. It searches for a series of frequent sets of items in the datasets. It builds association and correlations between item sets. It is the algorithm behind “You may also like” where you commonly saw in recommendation system.

A frequent item set is an item set whose support value is greater than a threshold value already specified. Like if A and B is a frequent item set then A and B should also be frequent items sets individually. Let’s better understand this with an example. We have transaction table as:-

|  |  |
| --- | --- |
| **Transaction ID** | **Item Set** |
| T1 | 1 3 4 |
| T2 | 2 3 5 |
| T3 | 1 2 3 5 |
| T4 | 2 5 |
| T5 | 1 3 5 |

Our first step is to build a list of item set of size 1 like {1} by using this transactional data. Here we take the minimum support count or value as **2/5**.

|  |  |
| --- | --- |
| **Item Set** | **Support** |
| {1} | 3/5 |
| {2} | 3/5 |
| {3} | 4/5 |
| {4} | 1/5 |
| {5} | 4/5 |

This table shows the support value of each item. You can see that the support value of item set {4} is 1/5 which is less than the given minimum support value. So, it will be eliminated.

|  |  |
| --- | --- |
| **Item Set** | **Support** |
| {1} | 3/5 |
| {2} | 3/5 |
| {3} | 4/5 |
| {5} | 4/5 |

So this is the final table for single item set. Now, the next step is to calculate the support of item set combined like {1, 2} or in pairs but not including item sets having {4} from here on.

|  |  |
| --- | --- |
| **Item Set** | **Support** |
| {1,2} | 1/5 |
| {1,3} | 3/5 |
| {1,5} | 2/5 |
| {2,3} | 2/5 |
| {2,5} | 3/5 |
| {3,5} | 3/5 |

Again we see that item set’s {1, 2} support value is less than the given minimum support value. So, this item set will be eliminated and not take in consideration from here on.

Now we will go for item set having three item set like {1, 2, 3}. Before calculating the support for three items of an item set we need to perform “Pruning”. Pruning is the method to check that after the combinations being made we decide {1, 2, 3} item set into subsets {1, 2, 3}, {1, 2}, {1, 3}, {2, 3} and from this sets we check if any subset’s support is less than the threshold support value. This is what frequent item set means.

|  |  |
| --- | --- |
| **Item Set** | **Is item set present in item sets of pairs like {a, b} ?** |
| {1, 2, 3}, {1, 2}, {1, 3}, {2, 3} | No - discard |
| {1, 2, 5}, {1, 2}, {1, 5}, {2, 5} | No - discard |
| {1, 3, 5}, {1, 5}, {1, 3}, {3, 5} | Yes - keep |
| {2, 3, 5}, {2, 3}, {2, 5}, {3, 5} | Yes - keep |

|  |  |
| --- | --- |
| **Item Set** | **Support** |
| {1, 3, 5} | 2/5 |
| {2, 3, 5} | 2/5 |

Thus we have two item sets for three items.

|  |  |
| --- | --- |
| **Item Set** | **Support** |
| {1, 2, 3, 5} | 2/5 |

For four items we have only one item set. The support value is less than given min. support value; so, we will stay with three items set.

Let’s assume that the minimum confidence value is 60%. For this we are going to generate all the non-empty subsets for each frequent item sets.

**For I = {1, 3, 5}, subsets are {1, 3}, {1, 5}, {3, 5}, {1}, {3}, {5}**

**For I = {2, 3, 5}, subsets are {2, 3}, {2, 5}, {2, 5}, {1}, {2}, {5}**

This rule states that for every subset S of I, the output of the rule gives something S gives I to S that recommends I of S. This is only possible if the support of I divided by support of S is greater than or equal to the minimum confidence value. Now applying this rules on item set {1, 3, 5} & {2, 3, 5} we get some rules. For the item set {1, 3, 5} it is done as:-

|  |  |
| --- | --- |
| **Rule ID** | **Rule selection or rejection based on minimum confidence value** |
| Rule 1 | {1, 3} => ({1, 3, 5} – {1, 3}) means 1 & 3 gives 5  Confidence = Support(1, 3, 5) / Support(1, 3) = 2/3 = 66.66% > 60%  Rule 1 is selected |
| Rule 2 | {1, 5} => ({1, 3, 5} – {1, 5}) means 1 & 5 gives 3  Confidence = Support(1, 3, 5) / Support(1, 5) = 2/2 = 100.00% > 60%  Rule 2 is selected |
| Rule 3 | {3, 5} => ({1, 3, 5} – {1, 3}) means 3 & 5 gives 1  Confidence = Support(1, 3, 5)/Support(3, 5) = 2/3 = 66.66% > 60%  Rule 3 is selected |
| Rule 4 | {1} => ({1, 3, 5} – {1}) means 1 gives 3 & 5  Confidence = Support(1, 3, 5) / Support(1) = 2/3 = 66.66% > 60%  Rule 4 is selected |
| Rule 5 | {3} => ({1, 3, 5} – {3}) means 3 gives 1 & 5  Confidence = Support(1, 3, 5) / Support(3) = 2/4 = 50.00% > 60%  Rule 5 is rejected |
| Rule 6 | {5} => ({1, 3, 5} – {5}) means 5 gives 1 & 3  Confidence = Support(1, 3, 5) / Support(5) = 2/4 = 50.00% > 60%  Rule 6 is rejected |