**Association Rule**

Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently a itemset occurs in a transaction. A typical example is Market Based Analysis.

Market Based Analysis is one of the key techniques used by large relations to show associations between items.It allows retailers to identify relationships between the items that people buy together frequently.

Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

Association rules finding is a rule-based machine learning technique used widely in recommender systems. If you have seen advertisements that are tailored according to your interests and preferences, chances are association rules were used to show you that specific advertisement. Association rule mining is principally the process of finding correlations between data points in a data set. The ‘rules’ here are the conditions used to specify the occurrence of a particular data point in a set given the occurrence of other data points. It finds which data points are the most correlated and are hence more likely to occur together. Essentially it learns rules of the form ‘If **X** occurs, **Y** will occur’. Each rule has some thresholds and parameters which specify how likely the rule is going to be true and other such properties. We then select the rules which are the best representative of the data and which are most suited for the task at hand.

To give a clearer picture, let’s say we have a dataset of the purchases made by the customers in a supermarket. We see that 70% percent of the customers who bought peanut butter also bought jelly (which is a popular combination used in sandwiches). Based on this dataset, we can then form an association between peanut butter and jelly such as ‘If a customer buys peanut butter, he will also buy jelly’. But this rule is not always true because we saw that only 70% of the customers followed this rule and the other 30% didn’t. Hence, we specify some properties of association rules that help us determine the efficacy of the rules. These are also called measures of interestingness since they help us determine which rules are interesting and are thereby useful.

This kind of association rules helps us determine several interesting correlations between sets in the dataset. Some of the associations are not-so-obvious ones and help significantly in market-oriented approaches, say for supermarkets to predict which items to bring in; or to show users advertisements specific to their tastes and preferences. One interesting and famous correlation found from association rule forming is the correlation between beer and diapers being bought together frequently in supermarkets. This seems counter-intuitive correlation was later explained by the fact that new fathers are tasked with shopping, while the mothers are left with the baby. As this example goes to show that this sort of rule forming can reveal associations that are not obvious and can significantly help markets in providing a better experience to the users.

## What Association Rule Mining Aims to Achieve?

Association Rule Mining is one of the ways to find patterns in data. It finds:

* features (dimensions) which occur together
* features (dimensions) which are “correlated”

What does the value of one feature tell us about the value of another feature? For example, people who buy diapers are likely to buy baby powder. Or we can rephrase the statement by saying: If (people buy diaper), then (they buy baby powder). Note the if, then rule. This does not necessarily mean that if people buy baby powder, they buy diaper. In General, we can say that if condition A tends to B it does not necessarily mean that B tends to A. Watch the directionality!

## When to use Association Rules

We can use Association Rules in any dataset where features take only two values i.e., 0/1. Some examples are listed below:

* **Market Basket Analysis**is a popular application of Association Rules.
* People who visit webpage X are likely to visit webpage Y
* People who have age-group [30,40] & income [>$100k] are likely to own home

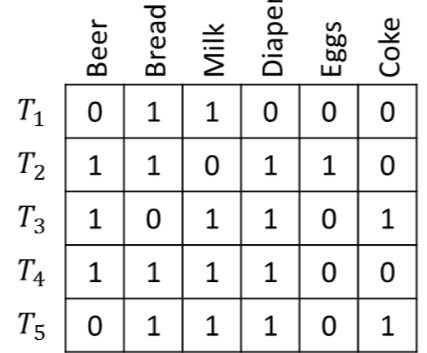
## Measures of Effectiveness of the Rule

The measures of effectiveness of the rule are as Follows:

* Support
* Confidence
* Lift
* Others: Affinity, Leverage



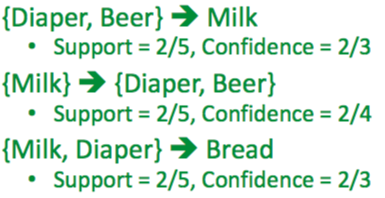
The above dataset can also be represented like this:



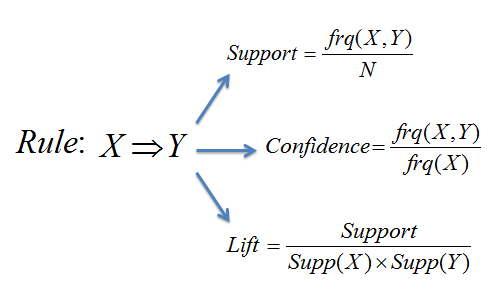
Support means how much historical data supports your rule and Confidence means how confident are we that the rule holds.

Support can be calculated as the fraction of rows containing both A and B or joint probability of A and B.

Among rows containing A, Confidence is the fraction of rows containing B or conditional probability of B given A.



**Lift** is the ratio Confidence is to Support. If the lift is < 1 then A and B are negatively correlated else positively correlated and if it is equal to 1 it is not correlated.



---------------------------------------------------------------------------

## What Is Association Rule Mining?

Refer ------<https://intellipaat.com/blog/data-science-apriori-algorithm/#What-Is-Association-Rule-Mining>

As mentioned before, the Apriori algorithm is used for the purpose of association rule mining. Now, what is association rule mining? Association rule mining is a technique to identify frequent patterns and associations among a set of items.

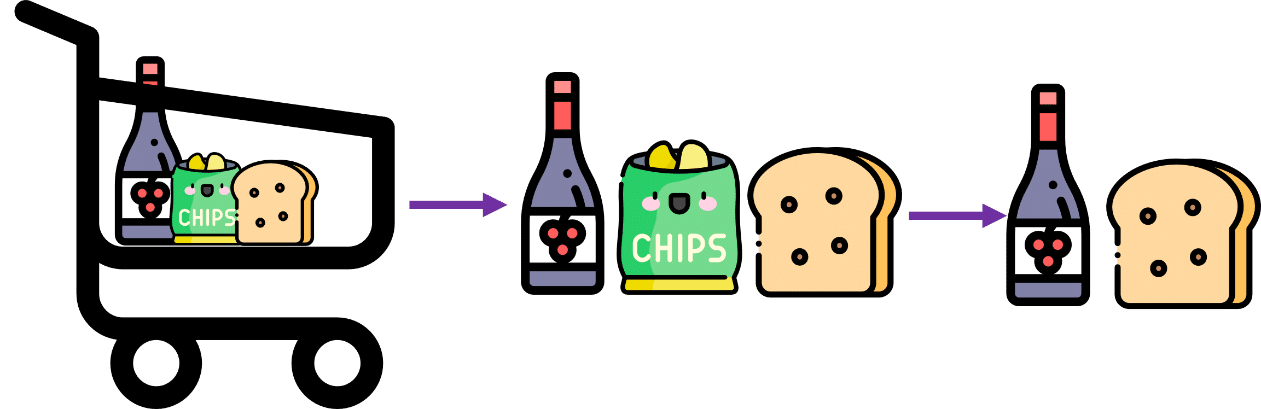
For example, understanding customer buying habits. By finding correlations and associations between different items that customers place in their ‘shopping basket,’ recurring patterns can be derived.

Say, Joshua goes to buy a bottle of wine from the supermarket. He also grabs a couple of chips as well. The manager there analyses that, not only Joshua, people often tend to buy wine and chips together. After finding out the pattern, the manager starts to arrange these items together and notices an increase in sales.

This process of identifying an association between products/items is called association rule mining. To implement association rule mining, many algorithms have been developed. Apriori algorithm is one of the most popular and arguably the most efficient algorithms among them.

## What Is an Apriori Algorithm?

Apriori algorithm assumes that any subset of a frequent itemset must be frequent.



## How Does the Apriori Algorithm Work?

The key concept in the Apriori algorithm is that it assumes all subsets of a frequent itemset to be frequent. Similarly, for any infrequent itemset, all its supersets must also be infrequent.

Let us try and understand the working of an Apriori algorithm with the help of a very famous business scenario, market basket analysis.

Here is a dataset consisting of six transactions in an hour. Each transaction is a combination of 0s and 1s, where 0 represents the absence of an item and 1 represents the presence of it.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Transaction ID | Wine | Chips | Bread | Milk |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 0 | 1 | 1 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 1 | 1 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |

We can find multiple rules from this scenario. For example, in a transaction of wine, chips, and bread, if wine and chips are bought, then customers also buy bread.

{wine, chips} => {bread}

In order to select the interesting rules out of multiple possible rules from this small business scenario, we will be using the following measures:

* Support
* Confidence
* List
* Conviction

### Support

Support of the item *x* is nothing but the ratio of the number of transactions in which the item *x* appears to the total number of transactions.

i.e.,

Support(wine) =  https://intellipaat.com/blog/wp-content/uploads/2019/05/Apiori3.png

Support(wine) = 4/6 = 0.6666

### Confidence

Confidence (*x* => *y*) signifies the likelihood of the item *y* being purchased when the item *x* is purchased. This method takes into account the popularity of the item *x*.

i.e.,

Conf({wine, chips} => {bread}) =https://intellipaat.com/blog/wp-content/uploads/2019/05/Apiori20.png

Conf({wine, chips} => {bread})= (2/6) / (3/6) = 0.667

### Lift

Lift (*x* => *y*) is nothing but the ‘interestingness’ or the likelihood of the item *y* being purchased when the item *x* is sold. Unlike confidence (*x* => *y*), this method takes into account the popularity of the item *y*.

i.e.,

lift ({wine, chips} => {bread}) = https://intellipaat.com/blog/wp-content/uploads/2019/05/Apiori20-1.png

lift ({wine, chips} => {bread}) = (2/6) / ((3/6) \* (4/6)) = 1

* Lift (x => y) = 1 means that there is no correlation within the itemset.
* Lift (x => y) > 1 means that there is a positive correlation within the itemset, i.e., products in the itemset, x and y, are more likely to be bought together.
* Lift (x => y) < 1 means that there is a negative correlation within the itemset, i.e., products in itemset, x and y, are unlikely to be bought together.

### Conviction

Conviction of a rule can be defined as follows:

conv(*x* => *y*) =

https://intellipaat.com/blog/wp-content/uploads/2019/05/Apriori24.png = https://intellipaat.com/blog/wp-content/uploads/2019/05/Apiori25.png=1

Its value range is [0, +∞].

* Conv(x => y) = 1 means that x has no relation with y.
* Greater the conviction higher the interest in the rule.

Now that we know the methods to find out the interesting rules, let us go back to the example. Before we get started, let us fix the support threshold to 50 percent.

### Step 1: Create a frequency table of all the items that occur in all transactions

|  |  |
| --- | --- |
| **Item** | **Frequency** |
| Wine | 4 |
| Chips | 4 |
| Bread | 4 |
| Milk | 5 |

### Step 2: Find the significant items based on the support threshold

Support threshold = 3

|  |  |
| --- | --- |
| **Item** | **Frequency** |
| Wine | 4 |
| Chips | 4 |
| Bread | 4 |
| Milk | 5 |

### Step 3: From the significant items, make possible pairs irrespective of the order

|  |  |
| --- | --- |
| **Item** | **Frequency** |
| Wine, Chips | 3 |
| Wine, Bread | 3 |
| Wine, Milk | 4 |
| Chips, Bread | 2 |
| Chips, Milk | 3 |
| Bread, Milk | 4 |

### Step 4: Again, find the significant items based on the support threshold

|  |  |
| --- | --- |
| **Item** | **Frequency** |
| Wine, Milk | 4 |
| Bread, Milk | 4 |

### Step 5: Now, make a set of three items that are bought together based on the significant items from Step 4

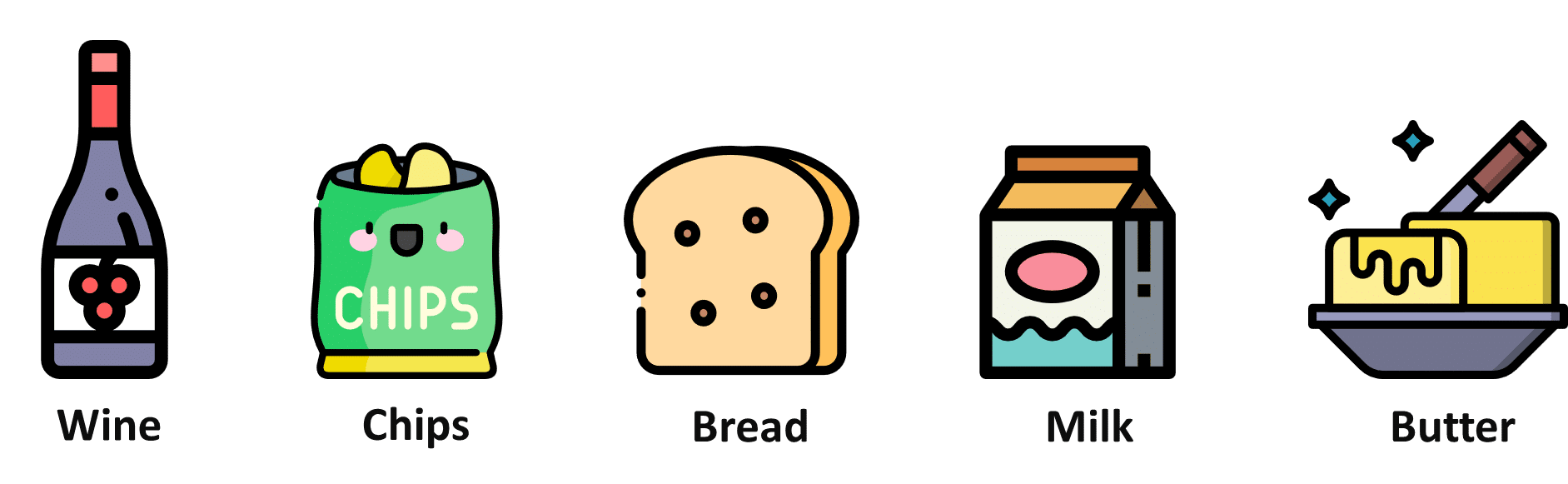
|  |  |
| --- | --- |
| **Item** | **Frequency** |
| Wine, Bread, Milk | 3 |

**{Wine, Bread, Milk}** is the only significant itemset we have got from the given data. But in real-world scenarios, we would have dozens of items to build rules from. Then, we might have to make four/five-pair itemsets.

## Hands-on: Apriori Algorithm in Python- Market Basket Analysis

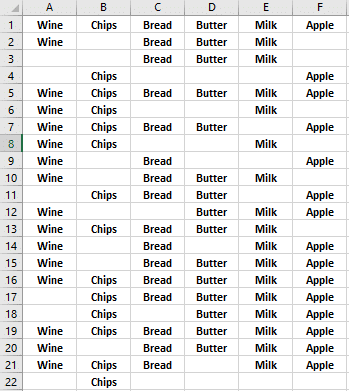
### Problem Statement

The manager of a retail store is trying to find out an association rule between six items, to figure out which items are more often bought together so that he can keep the items together in order to increase sales.



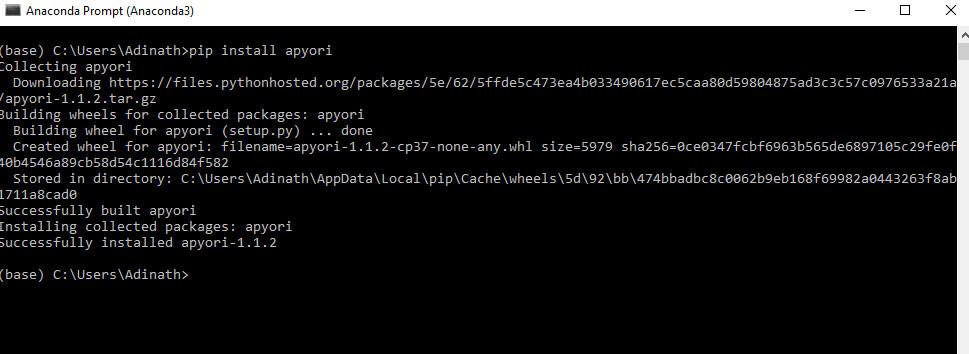
### Dataset

Below is the transaction data from Day 1. This dataset contains 6 items and 22 transaction records.



How to install apyori?

* Open anaconda prompt
* Pip install apyori



## Market Basket Analysis Implementation with in Python

With the help of **apyori** package, we will be implementing the Apriori algorithm in order to help the manager in market basket analysis.



### ****Step 1: Import the libraries****

#Importing the required datasets

import numpy as np

import pandas as pd

from apyori import apriori

### ****Step 2: Load the dataset****

#loading the dataset

basket\_data = pd.read\_csv("C:/Users/Adinath/Desktop/Data\_Science/Assignments/Association\_Rules/basket.csv", header = None)

#having a glipms at dataset

basket\_data

### ****Step 3: Have a glance at the records****



### ****Step 4: Look at the shape****

basket\_data = pd.read\_csv('C:/Users/Adinath/Desktop/Data\_Science/Assignments/Association\_Rules/basket.csv', delimiter=',', nrows = None) # Here nrows is number of rows we want to read, None=all rows, we can give any number here

basket\_data.dataframeName = 'groceries - groceries.csv'

nRow, nCol = basket\_data.shape

print(f'There are {nRow} rows and {nCol} columns')

### ****Step 5:**** ****Convert Pandas DataFrame into a list of lists****

#Converting panda's dataframe into a list of lists

records = []

for i in range (0, 21):

records.append([str(basket\_data.values[i,j]) for j in range (0, 6)])

### ****Step 6: Build the Apriori model****

#Building the first apripori model

association\_rules = apriori(records, min\_support=0.50, min\_confidence=0.7, min\_lift=1.2, min\_length=2)

association\_rules = list(association\_rules)

### ****Step 7: Print out the number of rules****

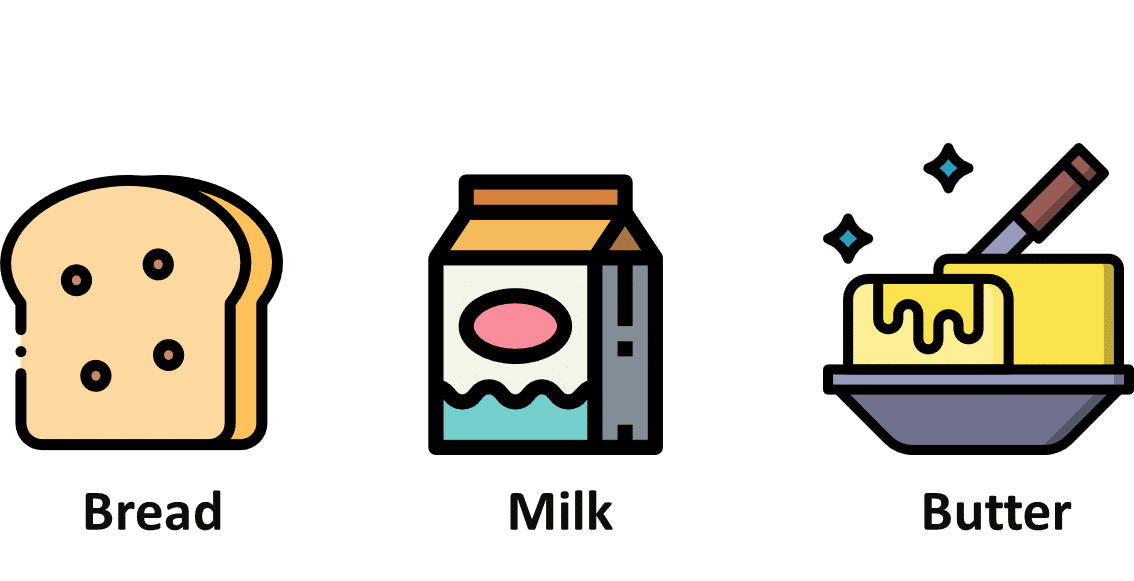
print(len(association\_rules))

o/p:=1

### ****Step 8: Have a glance at the rule****

print(association\_rules)

o/p:- [RelationRecord(items=frozenset({'Butter', 'Milk', 'Bread'}), support=0.5, ordered\_statistics=[OrderedStatistic(items\_base=frozenset({'Butter'}), items\_add=frozenset({'Milk', 'Bread'}), confidence=0.7333333333333334, lift=1.241025641025641), OrderedStatistic(items\_base=frozenset({'Milk', 'Bread'}), items\_add=frozenset({'Butter'}), confidence=0.8461538461538461, lift=1.241025641025641)])]



The support value for the first rule is 0.5. This number is calculated by dividing the number of transactions containing ‘Milk,’ ‘Bread,’ and ‘Butter’ by the total number of transactions.

The confidence level for the rule is 0.846, which shows that out of all the transactions that contain both “Milk” and “Bread”, 84.6 percent contain ‘Butter’ too.

The lift of 1.241 tells us that ‘Butter’ is 1.241 times more likely to be bought by the customers who buy both ‘Milk’ and ‘Butter’ compared to the default likelihood sale of ‘Butter.’

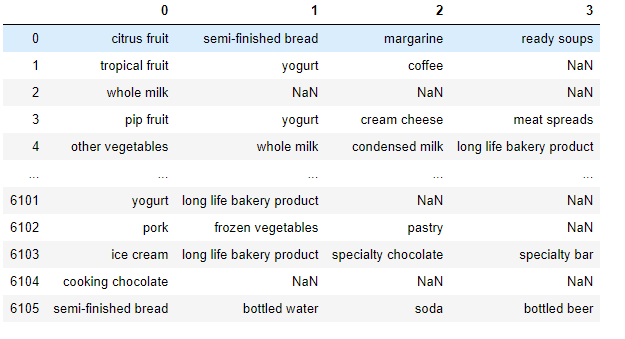
Groceris data

#loading the dataset

groceries\_data = pd.read\_csv("C:/Users/Adinath/Desktop/Data\_Science/Assignments/Association\_Rules/groceries.csv",error\_bad\_lines=False, header = None)

#having a glipms at dataset

groceries\_data



6106 rows × 4 columns

#Converting panda's dataframe into a list of lists

records = []

for i in range (0, 6106):

records.append([str(groceries\_data.values[i,j]) for j in range (0, 4)])

#Building the first apripori model

association\_rules = apriori(records, min\_support=0.50, min\_confidence=0.7, min\_lift=1, min\_length=2)

association\_rules = list(association\_rules)

print(len(association\_rules))

print(association\_rules)

o/p:- #Building the first apripori model

1

[RelationRecord(items=frozenset({'nan'}), support=0.8354077956108745, ordered\_statistics=[OrderedStatistic(items\_base=frozenset(), items\_add=frozenset({'nan'}), confidence=0.8354077956108745, lift=1.0)])]

In [44]:

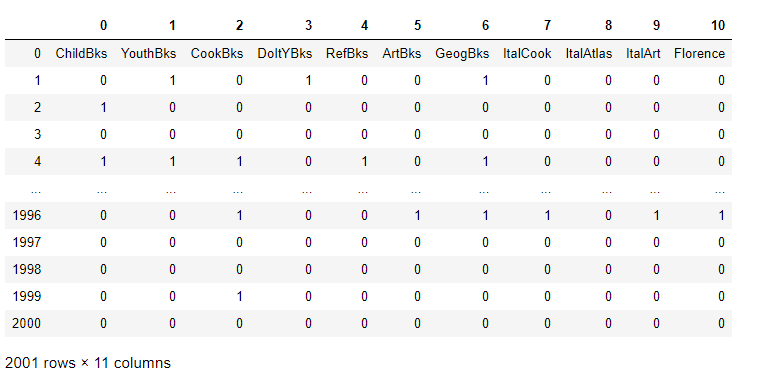
BOOK

#loading the dataset

book\_data = pd.read\_csv("C:/Users/Adinath/Desktop/Data\_Science/Assignments/Association\_Rules/book.csv", header = None)

#having a glipms at dataset

book\_data



#Building the first apripori model

association\_rules = apriori(records, min\_support=0.50, min\_confidence=0.7, min\_lift=1, min\_length=4)

association\_rules = list(association\_rules)

print(len(association\_rules))

print(association\_rules)

o/p:-

3

[RelationRecord(items=frozenset({'0'}), support=0.9985007496251874, ordered\_statistics=[OrderedStatistic(items\_base=frozenset(), items\_add=frozenset({'0'}), confidence=0.9985007496251874, lift=1.0)]), RelationRecord(items=frozenset({'1'}), support=0.7891054472763618, ordered\_statistics=[OrderedStatistic(items\_base=frozenset(), items\_add=frozenset({'1'}), confidence=0.7891054472763618, lift=1.0)]), RelationRecord(items=frozenset({'0', '1'}), support=0.7881059470264867, ordered\_statistics=[OrderedStatistic(items\_base=frozenset(), items\_add=frozenset({'0', '1'}), confidence=0.7881059470264867, lift=1.0), OrderedStatistic(items\_base=frozenset({'0'}), items\_add=frozenset({'1'}), confidence=0.7892892892892892, lift=1.0002329752171424), OrderedStatistic(items\_base=frozenset({'1'}), items\_add=frozenset({'0'}), confidence=0.9987333755541482, lift=1.0002329752171424)])]