**Theory:**

Has it ever happened that you’re out to buy something, and you end up buying a lot more than you planned? It’s a Phenomenon known as **Impulsive Buying** and Big Retailers take advantage of [Machine Learning](https://www.edureka.co/machine-learning-certification-training) and**Apriori Algorithm** and make sure that we tend to buy more.

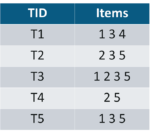
Apriori algorithm uses frequent itemsets to generate association rules. It is based on the concept that a subset of a frequent itemset must also be a frequent itemset. Frequent Itemset is an itemset whose support value is greater than a threshold value(support).

**Apriori Algorithm**

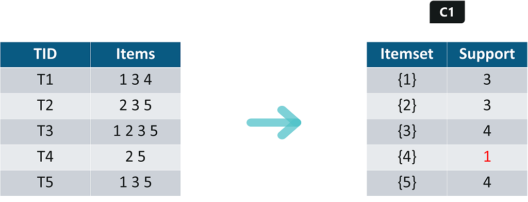
Apriori algorithm uses frequent itemsets to generate association rules. It is based on the concept that a subset of a frequent itemset must also be a frequent itemset. Frequent Itemset is an itemset whose support value is greater than a threshold value(support).



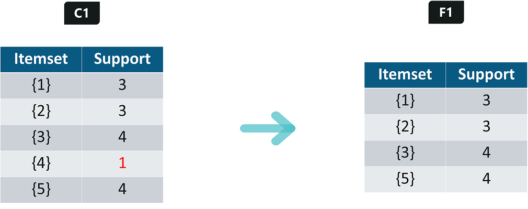
Let’s say we have the following data of a store.



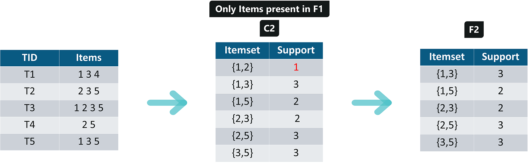
**Iteration 1:** Let’s assume the support value is 2 and create the item sets of the size of 1 and calculate their support values.



As you can see here, item 4 has a support value of 1 which is less than the min support value. So we are going to **discard {4}** in the upcoming iterations. We have the final Table F1.

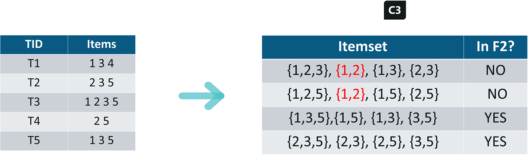


**Iteration 2:** Next we will create itemsets of size 2 and calculate their support values. All the combinations of items set in F1 are used in this iteration.



Itemsets having Support less than 2 are eliminated again. In this case **{1,2}.**Now, Let’s understand what is pruning and how it makes Apriori one of the best algorithm for finding frequent itemsets.

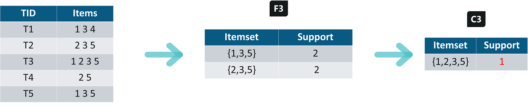
**Pruning:** We are going to divide the itemsets in C3 into subsets and eliminate the subsets that are having a support value less than 2.



**Iteration 3:** We will discard **{1,2,3}** and **{1,2,5}**as they both contain **{1,2}.**This is the main highlight of the Apriori Algorithm.



**Iteration 4:** Using sets of F3 we will create C4.



Since the Support of this itemset is less than 2, we will stop here and the final itemset we will have is F3.  
**Note:** Till now we haven’t calculated the confidence values yet.

With F3 we get the following itemsets:

**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}, {3,5}, {2}, {3}, {5}

**Applying Rules:** We will create rules and apply them on itemset F3. Now let’s assume a minimum confidence value is **60%.**

For every subsets S of I, you output the rule

* S –> (I-S) (means S recommends I-S)
* if **support(I) / support(S) >= min\_conf value**

**{1,3,5}**

**Rule 1:** {1,3} –> ({1,3,5} – {1,3}) means 1 & 3 –> 5

Confidence = support(1,3,5)/support(1,3) = 2/3 = **66.66%** **> 60%**

Hence Rule 1 is **Selected**

**Rule 2:** {1,5} –> ({1,3,5} – {1,5}) means 1 & 5 –> 3

Confidence = support(1,3,5)/support(1,5) = 2/2 = **100%** **> 60%**

Rule 2 is **Selected**

**Rule 3:** {3,5} –> ({1,3,5} – {3,5}) means 3 & 5 –> 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 –> 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 –> 1 & 5

Confidence = support(1,3,5)/support(3) = 2/4 = **50% <60%**

Rule 5 is **Rejected**

**Rule 6:** {5} –> ({1,3,5} – {5}) means 5 –> 1 & 3

Confidence = support(1,3,5)/support(5) = 2/4 = 50% < 60%

Rule 6 is **Rejected**

This is how you create rules in Apriori Algorithm and the same steps can be implemented for the itemset **{2,3,5}.**Try it for yourself and see which rules are accepted and which are rejected.

**Program:**

import pandas as pd

from mlxtend.frequent\_patterns import apriori

from mlxtend.frequent\_patterns import association\_rules

def encode\_units(x):

    if x <= 0:

        return 0

    if x >= 1:

        return 1

df = pd.read\_excel('D:\PROGRAMS\Online\_Retail.xlsx')

df.head()

df['Description'] = df['Description'].str.strip()

df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)

df['InvoiceNo'] = df['InvoiceNo'].astype('str')

df = df[~df['InvoiceNo'].str.contains('C')]

df

basket = (df[df['Country'] =="France"]

          .groupby(['InvoiceNo', 'Description'])['Quantity']

          .sum().unstack().reset\_index().fillna(0)

          .set\_index('InvoiceNo'))

basket

basket\_sets = basket.applymap(encode\_units)

basket\_sets.drop('POSTAGE', inplace=True, axis=1)

basket\_sets

frequent\_itemsets = apriori(basket\_sets, min\_support=0.07, use\_colnames=True)

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1)

rules.head()

rules[ (rules['lift'] >= 6) & (rules['confidence'] >= 0.8) ]

**Screenshots:**

