*Market Basket Analysis*

**Description:**

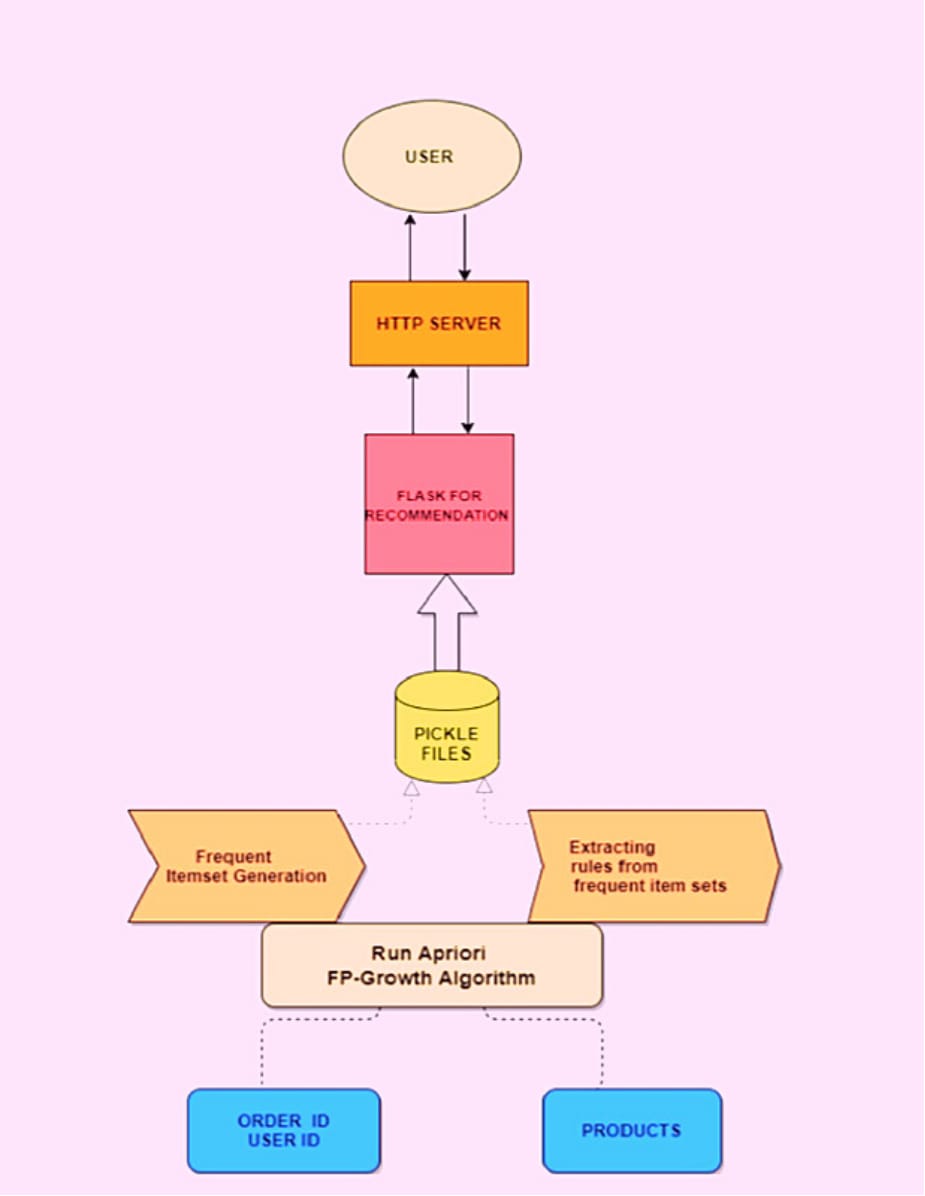
Market Basket analysis is a technique applied by retailers to understand customer’s shopping conduct from their stores. The result of the effective analysis may improve supplier’s profitability, quality of service and customer satisfaction. Instacart is a company that operates as a fast grocery delivery service in America. The purpose of this project is to make use of anonymized data on customers’ transactional orders to focus on descriptive analysis on the customer purchase patterns, items which are bought together and units that are highly purchased from the store to facilitate reordering and maintaining adequate product stock. It can be done by examine the available data in such way that frequent item set can be found and can be analyze to define an association rule. One of the algorithms which helps in finding association rule for frequent item set and to identify the correlation is Apriori algorithm. The model of the apriori algorithm is developed to explore approaches for the application of the rules of association to recommender system. Minimum confidence and minimum support values used for mining rules are parameters of the foremost existence.

**Design Thinking:**

Technology has introduced new buying behaviours to customers, and the rise of e-commerce has provided new possibilities with retailers to reach those customers. Retailers are increasingly turning to data analytics to identify opportunities for improvement and assist customer product offerings. Market basket analysis has become one of the main techniques major retailers use to discover interconnections between products. It operates by looking for combinations of items which mostly occur together in transactions. It helps retailers to define relationships between the products that customers purchase, to put it another way. Based on the principle of strong rules, association rules are commonly used to analyse retail basket or transaction data and are intended to define strong rules found in transaction data using interesting steps.

**Design:**

Here is the constructed Flowchart for the Market Basket Insights,



In the current project, by applying association rules on Instacart transactional data of the

customers it does not extract preferences of the individual customer rather it does specify the

connections at product levels of each transaction for all the customers. There by, mined rules

are used for the recommendation system. Perhaps, the strong association rules are supportive

for the recommender system in the approach of suggestions of most likely corelated with

similar products. Large datasets have been used for the analysis of the pattern and in order to

overcome the memory and computing issues, analysis of the data has been done in Google

Collab with the usage of GPU while unstacking the tractional data of the customers.

Recommendation system can be split into four stages:

1. The collection and pre-processing of raw data;

2. Convert pre-processed data into an easily achievable form using the selected machine

learning method such as Apriori or FP Growth algorithm;

3. Create a model of learning (training) using transformed data;

4. Use the previously developed set of association rules to report recommendations to the

User

Python libraries, frameworks and modules are very simple and powerful as compared

with the old days. Python has replaced many of the industry's languages, one of the reasons for

this is its vast collection of libraries, and is one of the most popular programming languages

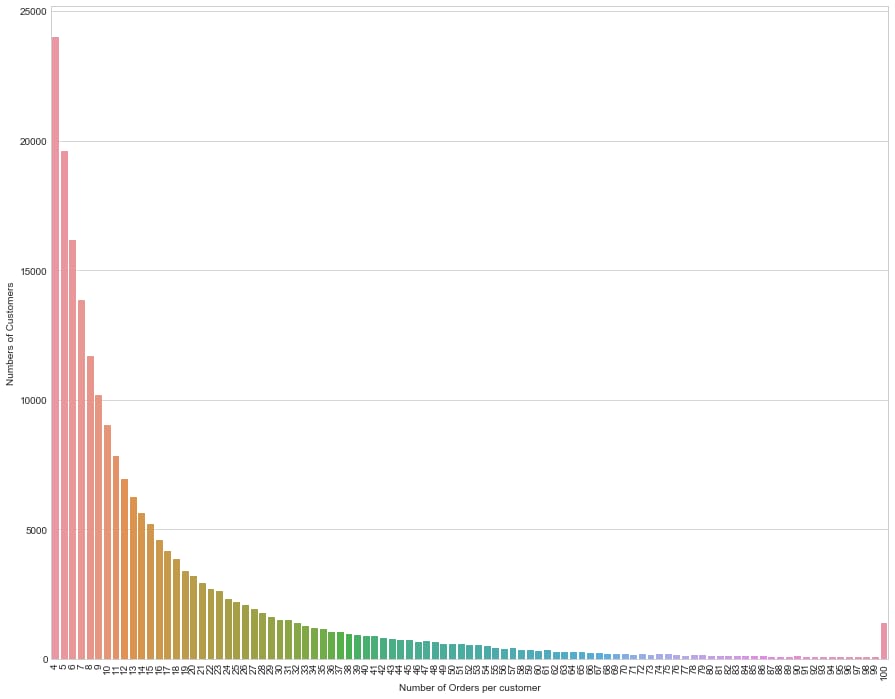
for this task today. Python libraries used in the project are:

* Pandas
* Numpy
* Matplotlib
* Seaborn
* MLxtend

Market basket analysis is a data mining technique used to discover associations and patterns in transactional data, such as customer purchase history. Data preparation is a crucial step in the process of conducting market basket analysis, as it involves getting the data ready for analysis. Here are the key steps involved in data preparation for market basket analysis:

* Data Collection
* Data Cleaning
* Data Transformation
* Data Encoding
* Support and Confidence Thresholds
* Apriori Algorithm (or other association rule mining algorithms)
* Rule Selection
* Post-Processing
* Visualization
* Interpretation

What are the minimum and maximum orders received from customers?



**Frequency Of Orders per customer**

Here, there are only 4 orders from 23,986 customers, only 5 orders from 19,590 customers. As

the number of customers orders increases, the number of customers ordering decreases.

Customers large percentage make 4 to 12 orders. When a company figures out a way to increase

the number of repeat customer orders, the sales will increase.

The Apriori algorithm works in two steps:

Prune and Join:

1.Generate all frequent item sets – A frequent item set is an item set that has transaction support

above minimum support.

2.Generate all confident association rules from frequent item sets – A confident association

rule is a rule with confidence above minimum confidence.

To apply Apriori algorithm on Instacart dataset, the Apriori class is applied that is imported

from the Apyori library.

• k-itemset is the itemset which contains: k element number.

• Lk refers to frequent sets of items with; k items.

• Ck corresponds with frequent sets of candidate items with elements;k

There are a variety of programming languages that can be used for the back end and front end of a market basket analysis application. Some popular choices include:

**Back end:**

Python

PySpark

MLlib

**Front end:**

JavaScript

The Apriori Algorithm is a basic algorithm proposed by Agrawal & Srikant in 1994 for the determination of the frequent itemset for Boolean association rules. The principles of Apriori state that “if an itemset is frequent, then all its subset items will be frequent”. If the support for the itemset is more than the support level, the itemset is “frequent”. The algorithm is based on the prediction of items, which move from the previous stage on a regular basis. The name derived from the term "prior". Apriori algorithm includes the type of association rules in data mining. The rule that states associations between multiple attributes is often called affinity analysis or market basket analysis.

The Apriori algorithm works in two steps:

Prune and Join:

1.Generate all frequent item sets – A frequent item set is an item set that has transaction support above minimum support.

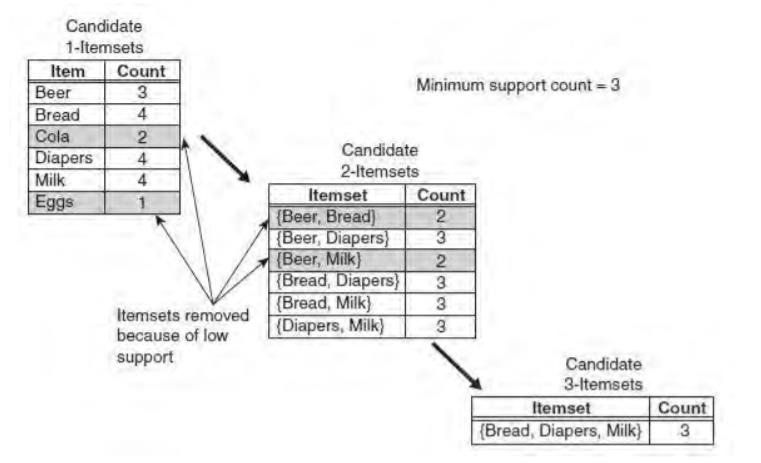
2.Generate all confident association rules from frequent item sets – A confident association rule is a rule with confidence above minimum confidence.

To apply Apriori algorithm on Instacart dataset, the Apriori class is applied that is imported from the Apyori library.

• k-itemset is the itemset which contains: k element number.

The Apriori function reduces the number of items to be searched to find frequent sets of items. This algorithm continues to identifying 2-itemsets using 1-itemsets, and 3-itemsets using 2-itemsets in an iterative way. This can be generalized as follows; frequent item sets

(k-1) elements are used to find frequent item candidates for k elements.



In the above instance, each item is initially considered as a 1-point candidate. Thereafter we count on their support and the itemset appearing less than minimum support count was discarded. As a consequence {Cola} and {Eggs} are removed. In the following iteration, candidate 2-itemset is generated with the help of frequent 1-itemset, because the Apriori principle ensures that all supersets of the rare 1-itemsets must be rare.

The number of candidate 2-itemsets generated by the algorithm is (4C2) = 6, because there are only four frequent 1-itemsets. The performance of the pruning strategy can be demonstrated by counting candidate generated item sets. A brute -force strategy to list al item sets (up to length 3) as candidates will give in 41 candidates.

**Frequent Pattern Growth Algorithm :**

The FP-Growth algorithm offers an alternate means of measuring a frequent item collection using an FP-Tree graphic data structure to compact transaction records. One can think of FP-Tree as turning the datasets into a graph format. Instead of the generation and check method used in the Apriori algorithm, FP-Growth generates the FP-Tree first, and uses this compact tree to produce the regular itemset. The FP-Growth algorithm's efficiency depends as to how much compression can be performed while generating the FP-Tree.The FP-Growth method transforms the problem of repeating the search of the minors and then combining the suffixes in the discovery of broad specific models. With the use of having slightly repetitive objects as a suffix it provides strong efficacy. This approach decreases search costs significantly.

**FP-Tree representation:**

A FP-tree is a compact data structure representing a collection of tree-shaped records. Every transaction is read out and sorted to an FP-tree path. This will come into force until all transactions are read out. The tree remains compact because the paths overlap by different transactions which have common subsets.

FP-Growth algorithm is the main execution mechanism [43];

1. Initially, scan the database and you can find items equal to and above the threshold value.

2. Support values for specific products are displayed in a size (large to small) order.

3. It then produces a tree with only roots.

4. The database is re-scanned for each sample;



A data collection of 5 transactions and five items can be seen in the diagram. The FP tree structures also appear in the figure after taking the first three transactions. Each node in the tree includes an item's label and also a tracker that represents the number of transactions that have taken the specified path. The way the FP-tree is generated is illustrated below:

1. The data is scanned at first to produce the support value for each item. Items which are not frequent are removed, but at the other side items which are frequent are organized in decreasing order. The figure above shows that a has been the most common item, then c, then d and ultimately e.

Shruthi Gurudath 2990078

2. The algorithm then crosses the data again for the FP-tree structure. The nodes a and b are generated after reading the first transaction {a, b}. The transaction in a tree is then generated from root- > a-> b. Now every node has its count value.

3. Then new nodes are created to represent b, c , and d when the second transaction is crossed {b , c, d}. Then a path is formed by the connection of the b , c and d nodes (root->b->c->d). Whereas the first two transactions involve b, these will not connect because they have a separate predecessor.

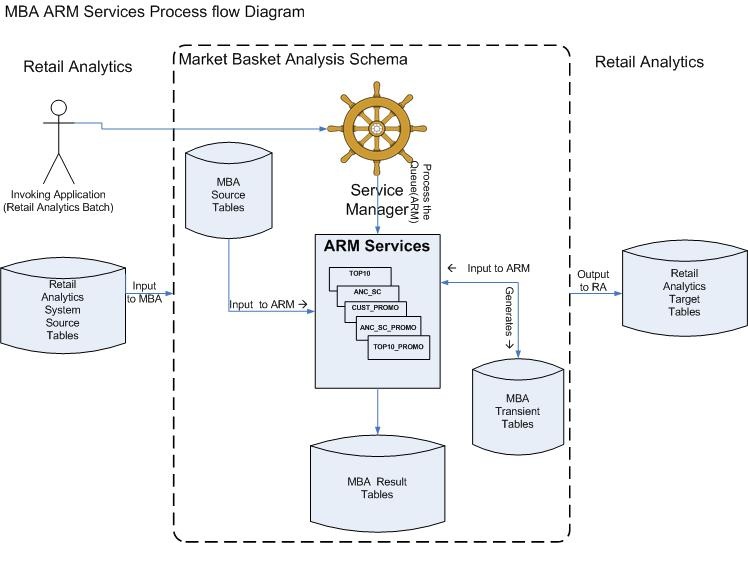
4. Then perhaps the third transaction {a, c, d , e} has an initially transacted common predecessor. As long as the item predecessor matches, the path overlaps.

5. The same process goes on until all data is put into the FP-tree.

Market basket insights involve examining the purchasing patterns of customers to identify relationships between different products or items they buy together. This analysis is particularly useful for businesses, especially in the retail and e-commerce industries, as it can help improve product recommendations, optimize store layouts, and plan marketing strategies. Here's how AI is used for market basket insights:

1. Data Collection: AI systems collect transaction data, such as point-of-sale data, e-commerce purchase history, or customer receipts. This data typically includes information about the items purchased and the transactions' timestamps.
2. Data Preprocessing: Raw transaction data may contain noise or inconsistencies. AI is used to clean and preprocess the data, ensuring that it is ready for analysis.
3. Association Rule Mining: AI algorithms, such as Apriori and FP-growth, are used to identify associations or relationships between items in customers' baskets. These algorithms find item sets that tend to appear together in transactions more frequently than expected by chance.
4. Support, Confidence, and Lift: AI calculates metrics like support, confidence, and lift for each association rule. These metrics help determine the significance and strength of the relationships between items in the market basket analysis.
   * Support: Measures the frequency of an itemset in the data.
   * Confidence: Measures how often the rule is true.
   * Lift: Measures how much more likely items are bought together than if they were bought independently.
5. Rule Generation: AI generates a set of association rules based on the chosen support, confidence, and lift thresholds. These rules provide insights into which items are frequently purchased together.
6. Visualization: AI can be used to create visualizations and reports to help businesses interpret the results of market basket analysis. Visualizations may include network diagrams, heatmaps, and item co-occurrence matrices.
7. Recommendations: AI-powered recommendation systems use the generated association rules to provide personalized product recommendations to customers. These recommendations can be displayed on e-commerce websites or in-store to encourage cross-selling and upselling.
8. Inventory and Supply Chain Management: Market basket insights can also be used to optimize inventory management and supply chain logistics. Businesses can adjust their stock levels and reorder strategies based on the items that tend to be purchased together.
9. Marketing Campaigns: Businesses can use market basket insights to design targeted marketing campaigns, including bundling related products or offering discounts on items frequently purchased together.
10. Store Layout Optimization: AI can help retailers optimize the layout of physical stores based on the insights from market basket analysis. Placing related items near each other can increase sales.

Flow Diagram:

****

**Step 1 – Importing required libraries**

We will be using the **mlxtend** Apriori library in Python, as shown below:

*//Import required libraries*

**import** pandas **as** pd

**from** mlxtend.frequent\_patterns **import** apriori

**from** mlxtend.frequent\_patterns **import** association\_rules

**Step 2 – Loading the data**

*//Load the data*

data = pd.read\_csv('data.csv', encoding= 'unicode\_escape')

**Step 3 – Cleaning the data**

*//Remove spaces from the description column*

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

*#Drop rows without invoice number*

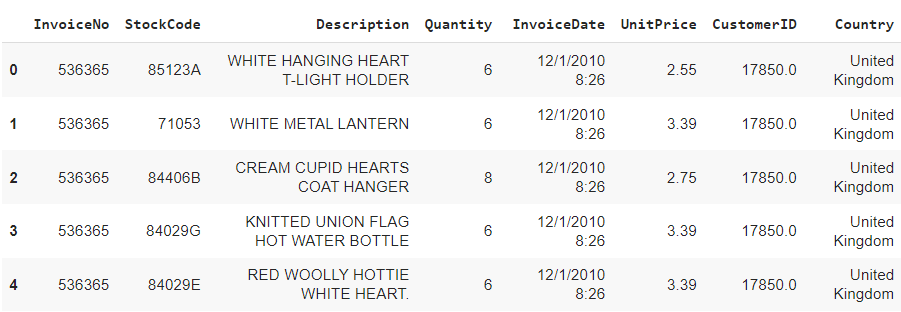
data.dropna(axis=0, subset=['InvoiceNo'], inplace=True)

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

*//Remove the credit transaction with invoice numbers containing 'C'*

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

data.head()



**Step 4 – Creating basket**

We are going to create a basket matrix by grouping multiple items within the same order and then unstack our DataFrame:

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

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

          .sum()

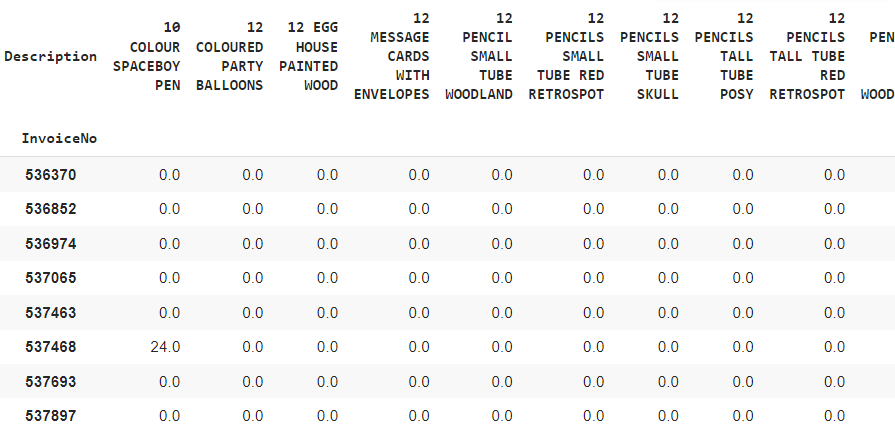
          .unstack()

          .reset\_index()

          .fillna(0)

          .set\_index('InvoiceNo'))

basket.head(10)



**Step 5 – Encoding**

We now need to encode the values in our matrix to 1’s and 0’s. We will do this in the following manner:

* Set the value to 0 if it is less than or equal to 0.
* Set the value to 1 if it is greater than or equal to 1.

**def** encode\_units(x):

**if** x <= 0:

**return** 0

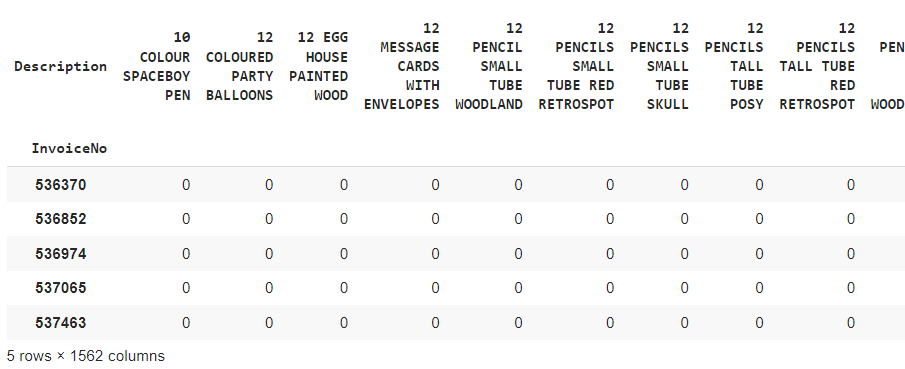
**if** x >= 1:

**return** 1

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

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

basket\_sets.head()



**Step 6 – Generating frequent item sets**

We will generate frequent item sets that have a support of at least 7%: //*Generate frequent itemsets*

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

**Step 7 – Generating association rules**

We will generate association rules with their corresponding support, confidence, and lift:

*//Generating the rules*

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

rules.head()

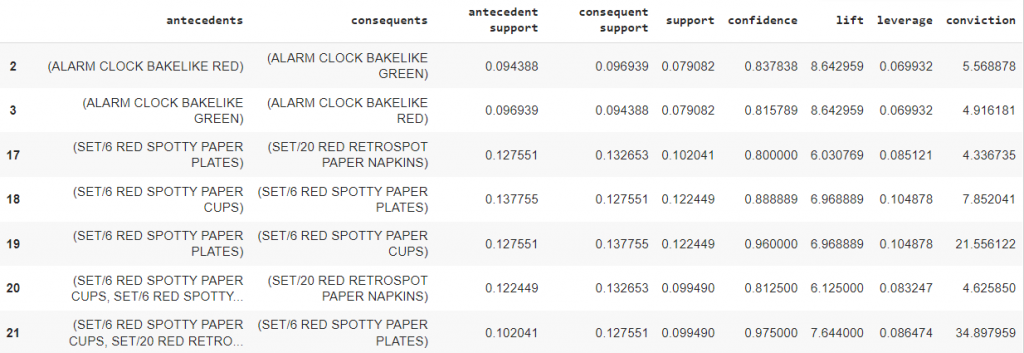
**Step 8 – Filtering rules with high Confidence and Lift**

As we already know, the greater the lift ratio, the more significant the association between the items. Also, the higher the confidence, the more reliable are the rules.

So, we are looking for rules with high confidence (>=0.8) and high lift (>=6). For this we filter the records as shown:

*//Filtering out the values with lift > = 6 and confidence > = 0.8*

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



So, we have our antecedents and consequent items along with their parameter values.

The result is giving us a lot of information about item grouping such as customers are 84% likely to buy the green alarm clock if they have already bought the red alarm clock!