

# Customer Shopping Basket Analysis for Small-Scale Grocery Stores and Vendors

February 18, 2022

-By Prachet Shah, A. Shraddha

## Step 1: Prototype Selection

### Abstract

Because of the Corona Pandemic and Lockdown, small-scale grocery shops and vendors have been affected a lot since the number of customers coming to them have reduced drastically and they have near to none profits.

So in my report, I want to apply the Association Rule Mining to predict and optimize the shopping experience for customers and selling experience of vendors and storeowners using Machine Learning and also provide a small-scale code implementation for it.

This technique is extensively used by large e-commerce stores and multi-chained supermarkets to boost their income and retain customers.

### 1.Problem Statement

To predict similarities in shopping habits and most bought combinations of items in grocery stores, so the vendor or store owner can know which items are sold the most and which items are bought frequently so that they can manage their inventory and come up with new schemes for increasing their sales and eventually profit.

It is an analyzing technique based on the idea that if we buy an item then we are bound to buy or not-buy a group (or single) item. For example, if a customer is buying bread then the chances of him/her buying jam is more.

## 2. Market/Customer/Business Need Assessment

There has been a huge downfall for Retail Shops and Vendors because of the unfortunate pandemic which has led many to buy things online instead of shops. So, it is extremely crucial for them to optimize their selling techniques by managing their inventory correctly by purchasing items which are in more demand and providing schemes on items which are generally grouped together to maximize their profits.

In addition, the epidemic has substantially altered client purchasing preferences. As a result of this technique, we hope to supply small businesses with beneficial data insights and revenue-generating opportunities.

## 3. Target Specifications and Characterization

1. **Boost of Sales:** Using this technique, vendors and shopkeepers can visualize which products or items are more profitable for them, so they can focus on which items to buy more for their inventory and which items to buy less according to customer needs so their management of loadout and finance is improved.
2. **Customer Retention:** With usage of analysis created by models, shopkeepers and vendors can group certain items together for easier shopping for customers and keep them happy and satisfied as they don't have to look for things bought in combination differently. For eg: keeping bread and butter closer to each other as they are bought in combination most of the times. For vendors, analysis aims to suggest to them, frequency of bought vegetables and fruits, so they can focus on that cultivation more to increase their sales and maximize their profits.

3. **Business Scheme:** Various schemes for promotion of business can be created by shopkeepers based on customer shopping habits so that customers can buy combinations of things for cheaper rates which will help in increasing profits. Eg: Discount on milk when customers buy cookies or cake powder as many buy them together.

## 4.External Search( Information and Data Analysis )

These are some of the sources I visited for more information and need for shopping pattern analysis of customers.

1. [Why Market Basket Analysis is Crucial to Gain a Winning Edge in the Retail Sector](#)
2. [Advantages of Market Basket Analysis in B2B Marketing](#)
3. [What is Market Basket analysis?](#)

I am going to use this [Dataset](#) for my code implementation for this report.

Dataset Description:

The dataset used in these models contains customers bought items. Each row corresponds to the item bought by one customer in one invoice. We have to find which items to be added in a buy one get one deal.

First import the basic libraries for data preprocessing:

## ▼ Eclat

### ▼ Importing the libraries

✓ [2] `!pip install apyori`

3s

Requirement already satisfied: apyori in /usr/local/lib/python3.7/dist-packages (1.1.2)

✓ [1] `import numpy as np`  
`import matplotlib.pyplot as plt`  
`import pandas as pd`

0s

## Let's now see more info on our dataset:

### ▼ Data Preprocessing

✓ [3] `dataset = pd.read_csv('Market_Basket_Optimisation.csv', header = None)`  
`transactions = []`  
`for i in range(0, 7501):`  
`transactions.append([str(dataset.values[i,j]) for j in range(0, 20)])`

0s

✓ [5] `dataset.head()`

0s

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice	frozen smoothie
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

✓ `print(f"Shape of Dataset is {dataset.shape}")`

0s

Shape of Dataset is (7501, 20)

✓ [4] dataset.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7501 entries, 0 to 7500
Data columns (total 20 columns):
#   Column  Non-Null Count  Dtype
---  -
0    0      7501 non-null     object
1    1      5747 non-null     object
2    2      4389 non-null     object
3    3      3345 non-null     object
4    4      2529 non-null     object
5    5      1864 non-null     object
6    6      1369 non-null     object
7    7      981 non-null      object
8    8      654 non-null      object
9    9      395 non-null      object
10   10     256 non-null      object
11   11     154 non-null      object
12   12     87 non-null       object
13   13     47 non-null       object
14   14     25 non-null       object
15   15     8 non-null        object
16   16     4 non-null        object
17   17     4 non-null        object
18   18     3 non-null        object
19   19     1 non-null        object
dtypes: object(20)
memory usage: 1.1+ MB
```

## Description and List of Transactions created for our Association Rule Mining Algorithm:

✓ [7] dataset.describe()

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
<b>count</b>	7501	5747	4389	3345	2529	1864	1369	981	654	395	256	154	87	47	25	8	4	4
<b>unique</b>	115	117	115	114	110	106	102	98	88	80	66	50	43	28	19	8	3	3
<b>top</b>	mineral water	mineral water	mineral water	mineral water	green tea	french fries	green tea	green tea	green tea	green tea	low fat yogurt	green tea	green tea	green tea	magazines	chocolate	frozen smoothie	protein bar
<b>freq</b>	577	484	375	201	153	107	96	67	57	31	22	15	8	4	3	1	2	2

✓ [8] print(list(transactions))

```
[[ 'shrimp', 'almonds', 'avocado', 'vegetables mix', 'green grapes', 'whole weat flour', 'yams', 'cottage cheese', 'energy drink', 'tomato juice',
```

## 5. Benchmarking

Lot of ecommerce stores like Amazon, Snapdeal, and Flipkart use these techniques to improvise their sales and to also create a smooth shopping experience for customers. Generally, Benchmarking involves comparing project processes and performance metrics to either industry best standards and practices or successful completed projects. For this there is a need to continuously search for implementation of better techniques which lead to better results or outputs.

## 6. Applicable Patents

1. [Method and system for researching product dynamics in market baskets in conjunction with aggregate market basket properties](#)
2. [Enhanced Market Basket Analysis](#)
3. [Method and apparatus for retail data mining using pair-wise co-occurrence consistency](#)

## 7. Applicable Regulations (Government and Environmental)

1. Data collection and Privacy of Regulations of Customers.
2. Government norms for Small Businesses and Street Vendors
3. Rules against False Marketing
4. Employment Schemes and laws created by government

## 8. Applicable Constraints

1. Lack of initial data to perform algorithms.
2. Convincing Shopkeepers and vendors to use this technique of selling over traditional means.
3. Lack of technical knowledge of vendors.

4. Rarely bought items will not be detected by algorithm, so it won't be generated as an output, so shopkeepers need to note which items are rarely bought and buy them in small quantities.
5. Need to continuously update and manage the data and model.

## 9. Business Opportunity

This technique of using Association Rule Mining to group Product Combinations and recommendations is extensively used by larger companies and it is still improving day by day. When small shop owners and vendors start using these techniques, they will not only improve their sales but they will also have an in-depth analysis of what things customers are buying and what they are not buying. That will also help them with maintaining their budget and which will eventually help them increase their reach and have growth in their business.

## 10. Concept Generation

This product requires the Machine Learning algorithm of Apriori to be written from scratch based on our needs and requirements. The optimization of the model is generally found out through tweaking in some hyperparameters. There are two methods to do this. If we want to have high speed but low lift and confidence, we can use the Eclat algorithm which is a derived algorithm from Apriori which only groups based on support parameters, but for overall high working efficiency we will be using the Apriori Algorithm here.

### Hyperparameters:

We have taken minimum support = 0.003 , minimum confidence = 20%, minimum lift = 3,

minimum length = 2 and maximum length = 2

To have better output and we found the results satisfactory.

Then we have results generated for the hyperparameters according to them .

Then we come up with the result of hyper parameters in the dataset with confidence in it.

## ▼ Training the Apriori model on the dataset

```
✓ [9] from apyori import apriori
    rules = apriori(transactions = transactions, min_support = 0.003, min_confidence = 0.2, min_lift = 3, min_length = 2, max_length = 2)
```

## ▼ Visualising the results

### ▼ Displaying the first results coming directly from the output of the apriori function

```
✓ [10] results = list(rules)
```

```
✓ [11] results
```

```
[RelationRecord(items=frozenset({'chicken', 'light cream'}), support=0.004532728969470737, ordered_statistics=[OrderedStatistic(items_base=frozenset({'chicken', 'light cream'}), support=0.004532728969470737, ordered_statistics=[]), OrderedStatistic(items_base=frozenset({'light cream'}), support=0.004532728969470737, ordered_statistics=[OrderedStatistic(items_base=frozenset({'light cream'}), support=0.004532728969470737, ordered_statistics=[])])]),
RelationRecord(items=frozenset({'escalope', 'mushroom cream sauce'}), support=0.005732568990801226, ordered_statistics=[OrderedStatistic(items_base=frozenset({'escalope', 'mushroom cream sauce'}), support=0.005732568990801226, ordered_statistics=[]), OrderedStatistic(items_base=frozenset({'mushroom cream sauce'}), support=0.005732568990801226, ordered_statistics=[OrderedStatistic(items_base=frozenset({'mushroom cream sauce'}), support=0.005732568990801226, ordered_statistics=[])])]),
RelationRecord(items=frozenset({'escalope', 'pasta'}), support=0.005865884548726837, ordered_statistics=[OrderedStatistic(items_base=frozenset({'escalope', 'pasta'}), support=0.005865884548726837, ordered_statistics=[]), OrderedStatistic(items_base=frozenset({'pasta'}), support=0.005865884548726837, ordered_statistics=[OrderedStatistic(items_base=frozenset({'pasta'}), support=0.005865884548726837, ordered_statistics=[])])]),
RelationRecord(items=frozenset({'honey', 'fromage blanc'}), support=0.003332888948140248, ordered_statistics=[OrderedStatistic(items_base=frozenset({'honey', 'fromage blanc'}), support=0.003332888948140248, ordered_statistics=[]), OrderedStatistic(items_base=frozenset({'fromage blanc'}), support=0.003332888948140248, ordered_statistics=[OrderedStatistic(items_base=frozenset({'fromage blanc'}), support=0.003332888948140248, ordered_statistics=[])])]),
RelationRecord(items=frozenset({'herb & pepper', 'ground beef'}), support=0.015997866951073192, ordered_statistics=[OrderedStatistic(items_base=frozenset({'herb & pepper', 'ground beef'}), support=0.015997866951073192, ordered_statistics=[]), OrderedStatistic(items_base=frozenset({'ground beef'}), support=0.015997866951073192, ordered_statistics=[OrderedStatistic(items_base=frozenset({'ground beef'}), support=0.015997866951073192, ordered_statistics=[])])]),
RelationRecord(items=frozenset({'tomato sauce', 'ground beef'}), support=0.005332622317024397, ordered_statistics=[OrderedStatistic(items_base=frozenset({'tomato sauce', 'ground beef'}), support=0.005332622317024397, ordered_statistics=[]), OrderedStatistic(items_base=frozenset({'ground beef'}), support=0.005332622317024397, ordered_statistics=[OrderedStatistic(items_base=frozenset({'ground beef'}), support=0.005332622317024397, ordered_statistics=[])])]),
RelationRecord(items=frozenset({'light cream', 'olive oil'}), support=0.003199573390214638, ordered_statistics=[OrderedStatistic(items_base=frozenset({'light cream', 'olive oil'}), support=0.003199573390214638, ordered_statistics=[]), OrderedStatistic(items_base=frozenset({'olive oil'}), support=0.003199573390214638, ordered_statistics=[OrderedStatistic(items_base=frozenset({'olive oil'}), support=0.003199573390214638, ordered_statistics=[])])]),
RelationRecord(items=frozenset({'whole wheat pasta', 'olive oil'}), support=0.007998933475536596, ordered_statistics=[OrderedStatistic(items_base=frozenset({'whole wheat pasta', 'olive oil'}), support=0.007998933475536596, ordered_statistics=[]), OrderedStatistic(items_base=frozenset({'olive oil'}), support=0.007998933475536596, ordered_statistics=[OrderedStatistic(items_base=frozenset({'olive oil'}), support=0.007998933475536596, ordered_statistics=[])])]),
RelationRecord(items=frozenset({'shrimp', 'pasta'}), support=0.005065991201173177, ordered_statistics=[OrderedStatistic(items_base=frozenset({'shrimp', 'pasta'}), support=0.005065991201173177, ordered_statistics=[]), OrderedStatistic(items_base=frozenset({'pasta'}), support=0.005065991201173177, ordered_statistics=[OrderedStatistic(items_base=frozenset({'pasta'}), support=0.005065991201173177, ordered_statistics=[])])]),
```

### ▼ Putting the results well organised into a Pandas DataFrame

```
✓ [12] def inspect(results):
    lhs      = [tuple(result[2][0][0])[0] for result in results]
    rhs      = [tuple(result[2][0][1])[0] for result in results]
    supports = [result[1] for result in results]
    confidences = [result[2][0][2] for result in results]
    lifts     = [result[2][0][3] for result in results]
    return list(zip(lhs, rhs, supports, confidences, lifts))
resultsinDataFrame = pd.DataFrame(inspect(results), columns = ['Left Hand Side', 'Right Hand Side', 'Support', 'Confidence', 'Lift'])
```

## 11. Concept Development

The required model can be launched by using the appropriate API (like Flask and Django) and its deployment can be done using Heroku for Flask API. The cloud services have to be chosen according to the need and budget of the customer.

Final Results:




## ▼ Displaying the results non sorted

✓ [13] resultsinDataFrame  
0s

	Left Hand Side	Right Hand Side	Support	Confidence	Lift
0	light cream	chicken	0.004533	0.290598	4.843951
1	mushroom cream sauce	escalope	0.005733	0.300699	3.790833
2	pasta	escalope	0.005866	0.372881	4.700812
3	fromage blanc	honey	0.003333	0.245098	5.164271
4	herb & pepper	ground beef	0.015998	0.323450	3.291994
5	tomato sauce	ground beef	0.005333	0.377358	3.840659
6	light cream	olive oil	0.003200	0.205128	3.114710
7	whole wheat pasta	olive oil	0.007999	0.271493	4.122410
8	pasta	shrimp	0.005066	0.322034	4.506672

## ▼ Displaying the results sorted by descending lifts

✓ 0s  resultsinDataFrame.nlargest(n = 10, columns = 'Lift')

	Left Hand Side	Right Hand Side	Support	Confidence	Lift
3	fromage blanc	honey	0.003333	0.245098	5.164271
0	light cream	chicken	0.004533	0.290598	4.843951
2	pasta	escalope	0.005866	0.372881	4.700812
8	pasta	shrimp	0.005066	0.322034	4.506672
7	whole wheat pasta	olive oil	0.007999	0.271493	4.122410
5	tomato sauce	ground beef	0.005333	0.377358	3.840659
1	mushroom cream sauce	escalope	0.005733	0.300699	3.790833
4	herb & pepper	ground beef	0.015998	0.323450	3.291994
6	light cream	olive oil	0.003200	0.205128	3.114710

## 12.Final Product Prototype/ Product Details

The final product provides service to operators about the most bought combinations of products for them to analyze customer shopping patterns and helps them manage their inventory and also create new strategies and schemes to increase their sales. The service implements the Market Basket Analysis, i.e Association Rule Mining technique on the dataset of transactions collected from the shopkeepers/vendors.

Some dynamics of the Apriori Algorithm used in this model and their meaning.

1. **Support:** It tells us about the combination of items bought together frequently. It gives the part of transactions that contain both A and B.

$$Support = \frac{freq(A, B)}{N}$$

2. **Confidence:** It tells us how frequently the items A and B are bought together, for the no. of times A is bought.

$$Confidence = \frac{freq(A, B)}{freq(A)}$$

3. **Lift:** It indicates the strength of a rule over the randomness of A and B being bought together. It basically measures the strength of any association rule.

$$Lift = \frac{Support}{Supp(A) \times Supp(B)}$$

## A) Feasibility

This project can be developed and deployed within a few years as SaaS( Software as a Service) for anyone to use.

## B) Viability

As the retail industry grows in India and the world, there will always be small businesses existing which can use this service to improvise on their sales and data warehousing techniques. So, it is viable to survive in the long-term future as well but improvements are necessary as new technologies emerge.

## C) Monetization

This service is directly monetizable as it can be directly released as a service on completion which can be used by businesses.

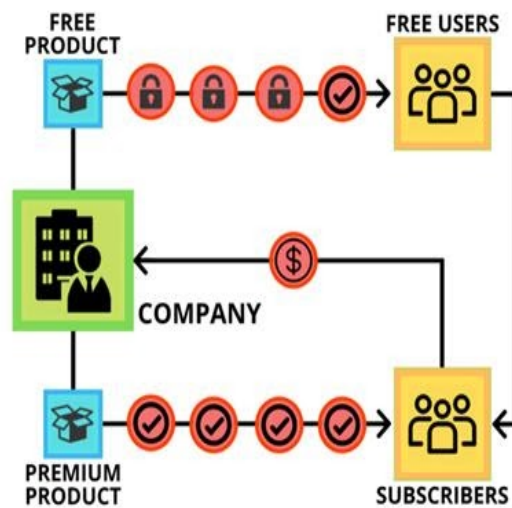
## Step 2: Prototype Development

Github Link: [Customer Shopping Basket Analysis for Small-Scale Grocery Stores and Vendors](#)

## Step 3: Business Modeling

For this service, it is beneficial to use a **Subscription Based Model**, where initially some features will be provided for free to engage customer retention and increase our customer count. Later it will be charged a subscription fee to use the service further for their business. In the subscription business model, customers pay a fixed amount of money on fixed time intervals to get access to the product or service provided by the company. The major problem is user conversion; how to convert the users into paid users.

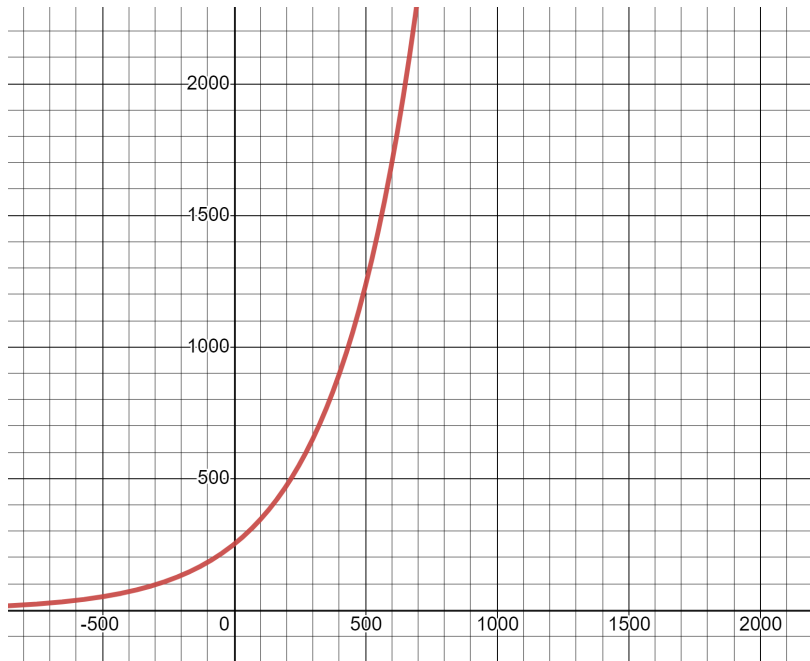
### SUBSCRIPTION BUSINESS MODEL



## Step 4: Financial Modeling

It can be directly launched into the retail market.

Let's consider our price of product = 250 for getting our graph



Financial Equation:

$$Y = X * (1 + r)^t$$

$$Y = (X) * (3.2)^t$$

$Y$  = Profit over time,  $X$  = Price of our Product,  $r$  = growth rate,  $t$  = time interval

$$1+r = 1 + 3.2\% = 1.032$$

## Conclusion

Market basket analysis is being used by an increasing number of companies to acquire beneficial insights about associations and hidden relationships. However, for small businesses, this extension is a fantastic opportunity to boost sales and help them develop and grow their business.