MARKET BASKET INSIGHTS

## TEAM MEMBER

**963521106027: DHANEESH A**

**PHASE 2**

INNOVATION

**TITLE**: **MARKET BASKET INSIGHTS**

**Abstract:**

Market basket analysis is a data mining technique that is used to identify patterns in customer purchases. It can be used to identify products that are frequently purchased together, as well as products that are rarely purchased together. This information can be used to develop marketing strategies, such as cross-selling and product placement.



**Problem Definition:**

In this phase, we can explore innovative techniques such as ensemble methods and deep learning architectures to improve the prediction system's accuracy and robustness.

**INNOVATION:**

Here we use Advanced association analysis techniques it can be used to explore market basket data in more depth. These techniques can help to identify more complex patterns, such as products that are frequently purchased together in sequences, as well as products that are purchased together in different customer segments.

This abstract will discuss the use of advanced association analysis techniques to explore market basket data. It will provide a brief overview of the different techniques, and discuss how they can be used to gain valuable insights into customer behaviour.

The abstract will also discuss some of the challenges associated with using advanced association analysis techniques, and how these challenges can be overcome. It will conclude by discussing the potential benefits of using advanced association analysis techniques for market basket insights.

Here are some specific examples of how advanced association analysis techniques can be used to explore market basket data:

* **Sequential pattern mining:** This technique can be used to identify products that are frequently purchased together in sequences. For example, a retailer might discover that customers who purchase diapers are also likely to purchase baby wipes and formula. This information could be used to develop promotions or product placement strategies.
* **Bi clustering**: This technique can be used to identify groups of products that are frequently purchased together by specific customer segments. For example, a retailer might discover that a group of young, urban customers are frequently purchasing coffee, laptops, and headphones together. This information could be used to develop targeted marketing campaigns.
* **Association rule mining with constraints**: This technique can be used to identify association rules that meet certain criteria. For example, a retailer might be interested in finding association rules that apply to customers who have spent more than a certain amount of money or who have purchased from a certain category of products.

Advanced association analysis techniques can provide valuable insights into customer behaviour. However, it is important to note that these techniques can be complex and require specialized knowledge to use effectively. Additionally, it is important to use high-quality data and to carefully consider the business context when interpreting the results of association analysis.

Overall, advanced association analysis techniques can be a powerful tool for exploring market basket data and gaining valuable insights into customer behaviour. By using these techniques, businesses can develop more effective marketing strategies and improve their bottom line.

**Dataset:**

**Dataset Link:**[**https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis**](https://www.kaggle.com/datasets/aslanahmedov/market-basket-analysis)

**Sample codes:**

# **CLUSTERING**

In [15]:

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

rfm\_scaled = scaler.fit\_transform(rfm\_data)

**Let's determine a cluster number**

In [16]:

linkcode

inertia = []

from sklearn.cluster import KMeans

for i **in** np.arange(1,6):

kmeans = KMeans(n\_clusters = i)

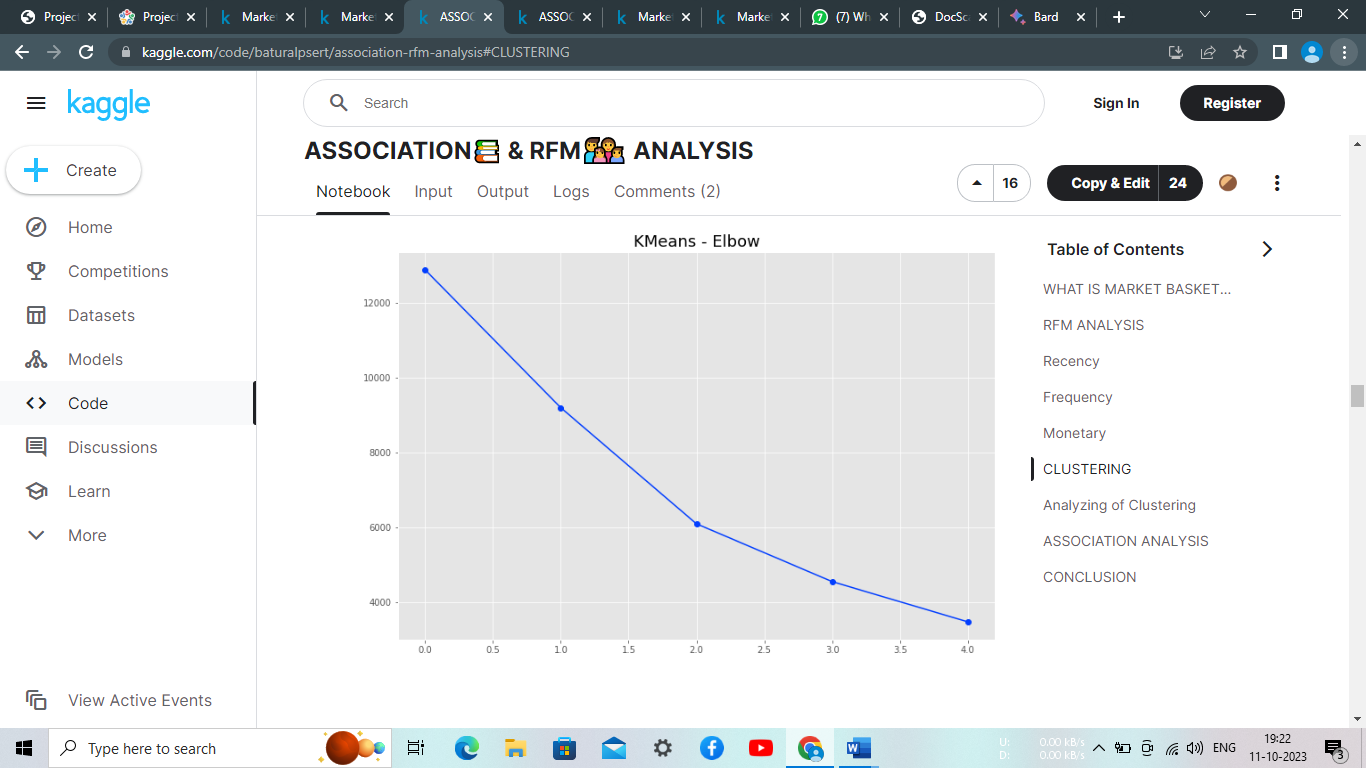
kmeans.fit(rfm\_scaled)

inertia.append(kmeans.inertia\_)

plt.figure(figsize = (12,8))

plt.plot(inertia, marker = "o")

plt.title("KMeans - Elbow", fontsize = 18);



**Best cluster number is 3. Let's create a clustering model.**

In [17]:

kmeans = KMeans(n\_clusters = 3)

kmeans.fit(rfm\_scaled)

rfm\_data["Cluster\_No"] = (kmeans.labels\_ + 1)

In [18]:

rfm\_data.head()

Out[18]:

|  | Recency | Frequency | Monetary | Cluster\_No |
| --- | --- | --- | --- | --- |
| CustomerID |  |  |  |  |
| 12346.0 | 347 | 1 | 77183.60 | 3 |
| 12347.0 | 61 | 7 | 4310.00 | 2 |
| 12349.0 | 40 | 1 | 1757.55 | 2 |
| 12350.0 | 332 | 1 | 334.40 | 1 |
| 12352.0 | 94 | 8 | 2506.04 | 2 |

# **Analyzing of Clustering**

In [19]:

rfm\_data.groupby(["Cluster\_No"])[["Recency", "Frequency", "Monetary"]].mean()

Out[19]:

|  | Recency | Frequency | Monetary |
| --- | --- | --- | --- |
| Cluster\_No |  |  |  |
| 1 | 281.745299 | 1.545299 | 495.484189 |
| 2 | 68.634429 | 4.797872 | 1913.384218 |
| 3 | 48.760000 | 58.960000 | 81979.682000 |

Hmm. Our model determine **3 clusters** that

* **Cluster 1** --> Customers who haven't been here in a long time. We need to do some discount for them. We can still turn them back.
* **Cluster 2** --> Middle-level customers.
* **Cluster 3** --> Premium customers. We don't want to lose them. They spend a lot of money for us, and their recency is good.

In [20]:

linkcode

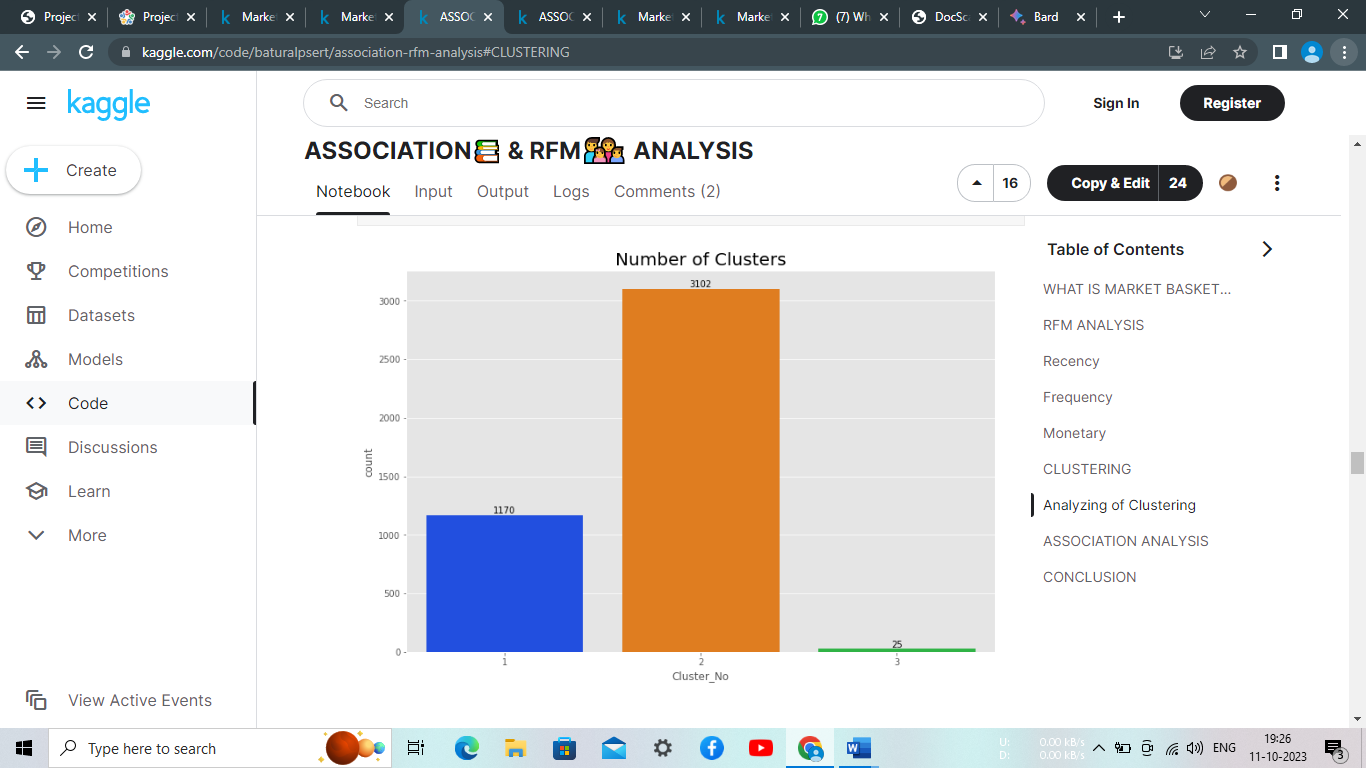
plt.figure(figsize = (12,8))

ax = sns.countplot(rfm\_data.Cluster\_No)

plt.title("Number of Clusters", fontsize = 20);

for bars **in** ax.containers:

ax.bar\_label(bars)



* **As we can see, only 25 people are premium customers,**
* **3102 people are middle-level customers**
* **1170 people are customers that we can turn back.**

**Let's visualize them with scatterplot.**

In [21]:

from sklearn.decomposition import PCA

pca = PCA(n\_components = 2)

pca = pca.fit\_transform(rfm\_scaled)

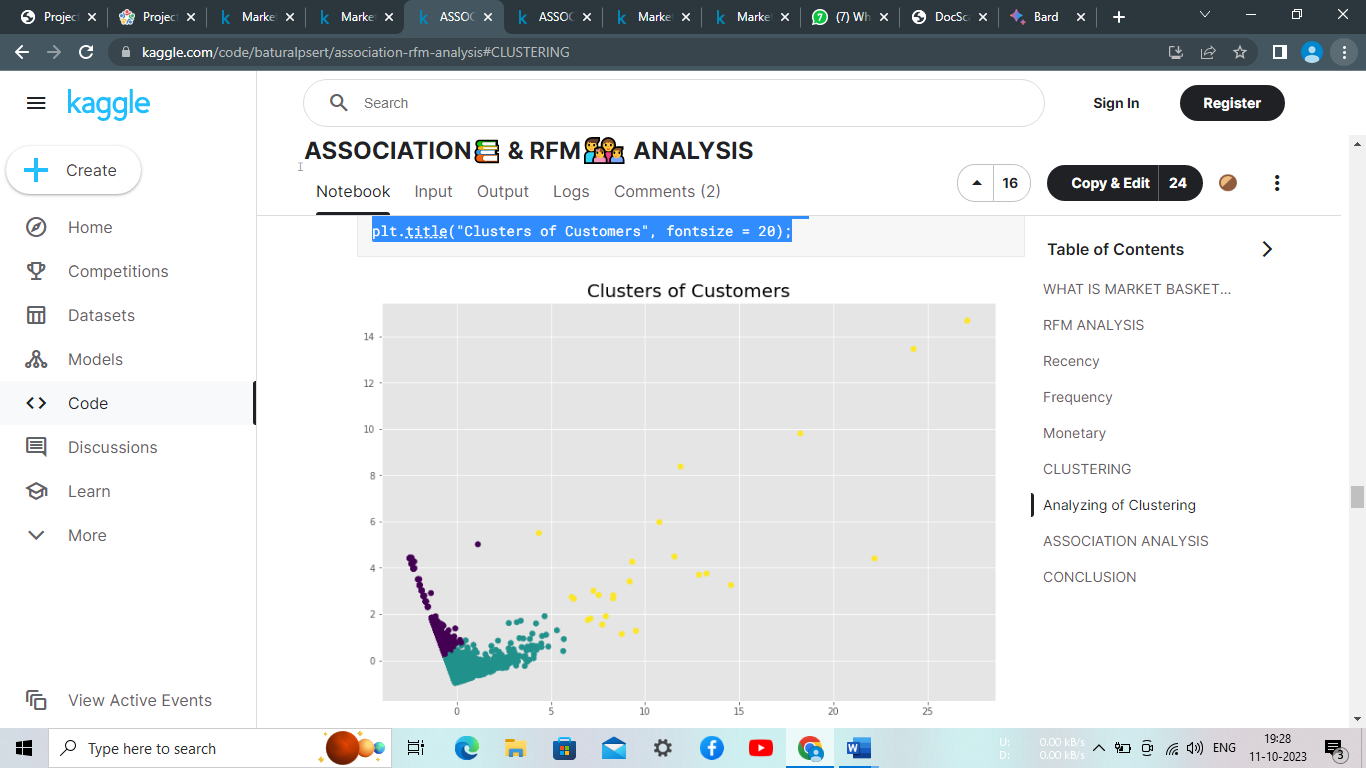
In [22]:

linkcode

plt.figure(figsize = (12,8))

plt.scatter(pca[:,0], pca[:,1], c = kmeans.labels\_)

plt.title("Clusters of Customers", fontsize = 20);



**Association rule mining with constrains:**

data\_apr = df.groupby(["BillNo", "Itemname"])[["Quantity"]].sum(

).unstack().reset\_index().fillna(0).set\_index("BillNo")

In [24]:

data\_apr.head()

Out[24]:

|  | Quantity | | | | | | | | | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Itemname | \*Boombox Ipod Classic | \*USB Office Mirror Ball | 10 COLOUR SPACEBOY PEN | 12 COLOURED PARTY BALLOONS | 12 DAISY PEGS IN WOOD BOX | 12 EGG HOUSE PAINTED WOOD | 12 HANGING EGGS HAND PAINTED | 12 IVORY ROSE PEG PLACE SETTINGS | 12 MESSAGE CARDS WITH ENVELOPES | 12 PENCIL SMALL TUBE WOODLAND | ... | wrongly coded 20713 | wrongly coded 23343 | wrongly coded-23343 | wrongly marked | wrongly marked 23343 | wrongly marked carton 22804 | wrongly marked. 23343 in box | wrongly sold (22719) barcode | wrongly sold as sets | wrongly sold sets |
| BillNo |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 536365 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 536366 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 536367 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 536368 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 536369 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

5 rows × 4185 columns

In [25]:

def num(x):

if x <= 0:

return 0

elif x >=1:

return 1

basket\_new = data\_apr.applymap(num)

In [26]:

basket\_new.nunique()

Out[26]:

Itemname

Quantity \*Boombox Ipod Classic 2

\*USB Office Mirror Ball 2

10 COLOUR SPACEBOY PEN 2

12 COLOURED PARTY BALLOONS 2

12 DAISY PEGS IN WOOD BOX 2

..

wrongly marked carton 22804 1

wrongly marked. 23343 in box 1

wrongly sold (22719) barcode 2

wrongly sold as sets 1

wrongly sold sets 1

Length: 4185, dtype: int64

In [27]:

*#!pip install mlxtend*

In [28]:

from mlxtend.frequent\_patterns import apriori

apr = apriori(basket\_new, min\_support = 0.02, use\_colnames = True)

apr.sort\_values(by = "support", ascending = False)

/opt/conda/lib/python3.7/site-packages/mlxtend/frequent\_patterns/fpcommon.py:115: DeprecationWarning: DataFrames with non-bool types result in worse computationalperformance and their support might be discontinued in the future.Please use a DataFrame with bool type

DeprecationWarning,

Out[28]:

|  | support | itemsets |
| --- | --- | --- |
| 263 | 0.108956 | ((Quantity, WHITE HANGING HEART T-LIGHT HOLDER)) |
| 97 | 0.102128 | ((Quantity, JUMBO BAG RED RETROSPOT)) |
| 190 | 0.094211 | ((Quantity, REGENCY CAKESTAND 3 TIER)) |
| 151 | 0.081940 | ((Quantity, PARTY BUNTING)) |
| 122 | 0.076249 | ((Quantity, LUNCH BAG RED RETROSPOT)) |
| ... | ... | ... |
| 274 | 0.020139 | ((Quantity, WOODEN UNION JACK BUNTING)) |
| 245 | 0.020139 | ((Quantity, STRAWBERRY SHOPPER BAG)) |
| 219 | 0.020139 | ((Quantity, SET OF 60 I LOVE LONDON CAKE CASES)) |
| 195 | 0.020040 | ((Quantity, RIBBON REEL STRIPES DESIGN)) |
| 354 | 0.020040 | ((Quantity, WOODEN PICTURE FRAME WHITE FINISH)... |

358 rows × 2 columns

In [29]:

from mlxtend.frequent\_patterns import association\_rules

end = association\_rules(apr, metric = "lift", min\_threshold = 1)

end.sort\_values(by = "confidence", ascending = False)

Out[29]:

|  | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 154 | ((Quantity, ROSES REGENCY TEACUP AND SAUCER), ... | ((Quantity, GREEN REGENCY TEACUP AND SAUCER)) | 0.028204 | 0.048243 | 0.025482 | 0.903509 | 18.728115 | 0.024122 | 9.863659 |
| 153 | ((Quantity, PINK REGENCY TEACUP AND SAUCER), (... | ((Quantity, ROSES REGENCY TEACUP AND SAUCER)) | 0.029936 | 0.050124 | 0.025482 | 0.851240 | 16.982778 | 0.023982 | 6.385280 |
| 25 | ((Quantity, PINK REGENCY TEACUP AND SAUCER)) | ((Quantity, GREEN REGENCY TEACUP AND SAUCER)) | 0.036418 | 0.048243 | 0.029936 | 0.822011 | 17.038810 | 0.028179 | 5.347273 |
| 160 | ((Quantity, JUMBO STORAGE BAG SUKI), (Quantity... | ((Quantity, JUMBO BAG RED RETROSPOT)) | 0.025433 | 0.102128 | 0.020386 | 0.801556 | 7.848573 | 0.017789 | 4.524572 |
| 139 | ((Quantity, PINK REGENCY TEACUP AND SAUCER)) | ((Quantity, ROSES REGENCY TEACUP AND SAUCER)) | 0.036418 | 0.050124 | 0.028204 | 0.774457 | 15.450905 | 0.026378 | 4.211500 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 70 | ((Quantity, JUMBO BAG RED RETROSPOT)) | ((Quantity, JUMBO STORAGE BAG SKULLS)) | 0.102128 | 0.034785 | 0.020435 | 0.200097 | 5.752430 | 0.016883 | 1.206665 |
| 161 | ((Quantity, JUMBO BAG RED RETROSPOT)) | ((Quantity, JUMBO STORAGE BAG SUKI), (Quantity... | 0.102128 | 0.025433 | 0.020386 | 0.199612 | 7.848573 | 0.017789 | 1.217619 |
| 42 | ((Quantity, JUMBO BAG RED RETROSPOT)) | ((Quantity, JUMBO BAG ALPHABET)) | 0.102128 | 0.043790 | 0.020336 | 0.199128 | 4.547316 | 0.015864 | 1.193961 |
| 131 | ((Quantity, WHITE HANGING HEART T-LIGHT HOLDER)) | ((Quantity, NATURAL SLATE HEART CHALKBOARD)) | 0.108956 | 0.060960 | 0.020336 | 0.186649 | 3.061823 | 0.013695 | 1.154532 |
| 149 | ((Quantity, WHITE HANGING HEART T-LIGHT HOLDER)) | ((Quantity, WOODEN PICTURE FRAME WHITE FINISH)) | 0.108956 | 0.054033 | 0.020040 | 0.183924 | 3.403936 | 0.014152 | 1.159165 |
|  |  |  |  |  |  |  |  |  |  |

164 rows × 9 columns

### **Associative Rule Mining with PyCaret:**

In [49]:

from pycaret.arules import \*

exp = setup(data=df, transaction\_id='BillNo', item\_id='Itemname')

| Description | Value |
| --- | --- |
| session\_id | 5711 |
| # Transactions | 19658 |
| # Items | 4052 |
| Ignore Items | None |

In [50]:

*# We have to greatly decrease the min\_support threshold as the number of rows are huge*

model1 = create\_model(threshold=0.5, min\_support=0.01, low\_memory=True)

In [51]:

model1

Out[51]:

|  | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | (BEADED CRYSTAL HEART PINK ON STICK) | (DOTCOM POSTAGE) | 0.0105 | 0.0360 | 0.0102 | 0.9757 | 27.0916 | 0.0098 | 39.7161 |
| 1 | (JAM MAKING SET PRINTED, SUKI SHOULDER BAG) | (DOTCOM POSTAGE) | 0.0111 | 0.0360 | 0.0106 | 0.9587 | 26.6193 | 0.0102 | 23.3498 |
| 2 | (HERB MARKER PARSLEY, HERB MARKER ROSEMARY) | (HERB MARKER THYME) | 0.0107 | 0.0119 | 0.0102 | 0.9524 | 80.3515 | 0.0100 | 20.7511 |
| 3 | (HERB MARKER THYME, HERB MARKER PARSLEY) | (HERB MARKER ROSEMARY) | 0.0107 | 0.0119 | 0.0102 | 0.9524 | 80.0081 | 0.0100 | 20.7500 |
| 4 | (REGENCY TEA PLATE PINK, REGENCY TEA PLATE ROSES) | (REGENCY TEA PLATE GREEN) | 0.0127 | 0.0181 | 0.0120 | 0.9478 | 52.4836 | 0.0118 | 18.8080 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 886 | (LUNCH BAG CARS BLUE, LUNCH BAG RED RETROSPOT) | (LUNCH BAG WOODLAND) | 0.0277 | 0.0511 | 0.0139 | 0.5009 | 9.8078 | 0.0125 | 1.9013 |
| 887 | (RED RETROSPOT CHARLOTTE BAG, STRAWBERRY CHARL... | (WOODLAND CHARLOTTE BAG, CHARLOTTE BAG PINK PO... | 0.0245 | 0.0199 | 0.0123 | 0.5000 | 25.0740 | 0.0118 | 1.9601 |
| 888 | (JUMBO BAG WOODLAND ANIMALS) | (JUMBO BAG PINK POLKADOT) | 0.0434 | 0.0616 | 0.0217 | 0.5000 | 8.1164 | 0.0190 | 1.8768 |
| 889 | (PICNIC BASKET WICKER SMALL) | (DOTCOM POSTAGE) | 0.0232 | 0.0360 | 0.0116 | 0.5000 | 13.8828 | 0.0108 | 1.9280 |
| 890 | (HAND WARMER RED LOVE HEART) | (HAND WARMER SCOTTY DOG DESIGN) | 0.0200 | 0.0277 | 0.0100 | 0.5000 | 18.0349 | 0.0095 | 1.9446 |

891 rows × 9 columns

# **CONCLUSION:**

**After looking these results it indicates the possibility that customers buying the X product will buy the Y product. We need to make a decision for them. Maybe in our website, when the customer click on first one, we need to show them the other item.**

**For example: When our customer clicks on PINK REGENCY TEACUP AND SAUCER, we need to show them GREEN REGENCY TEACUP AND SAUCER and maximize our profit.**