Market Basket Analysis

Introduction

In retailing, understanding your customer's behaviors and attitudes towards products is of utmost importance. Insights into these buying behaviors can be used to boost sales and improve the customer experience.

Market basket analysis is a data mining technique used by retailers to increase sales by better understanding customer purchasing patterns. It involves analyzing large data sets, such as purchase history, to reveal product groupings, as well as products that are likely to be purchased together. Market basket analysis provides a computational method for identifying associations between products, from which a strategy can be formed.

These insights help in restocking and warehousing decisions as well as customer recommendations and have been used to great effect in the success of companies such as Amazon.

Objectives

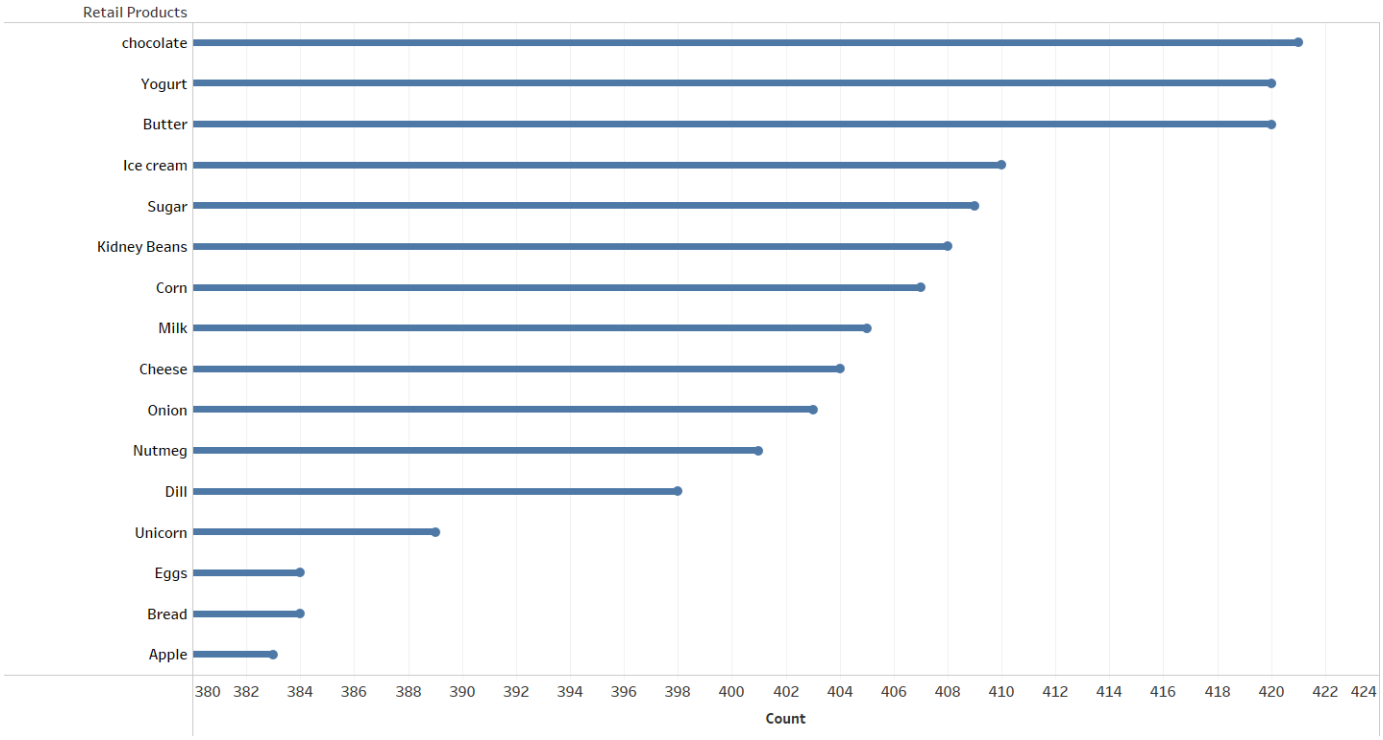
Our goal in this project is to analyze food product transactions and posit patterns or rules of customer buying behavior for use in similar item recommendations.

Dataset

The Dataset for this project is from Kaggle. It contains 999 entries and 17 columns of various purchased food items. The dataset was produced for the purpose of analyzing the products purchased in the same basket.

	Apple	Bread	Butter	Cheese	Corn	Dill	Eggs	Ice cream	Kidney Beans	Milk	Nutmeg	Onion	Sugar	Unicorn	Yogurt	chocolate
0	False	True	False	False	True	True	False	True	False	False	False	False	True	False	True	True
1	False	False	False	False	False	False	False	False	False	True	False	False	False	False	False	False
2	True	False	True	False	False	True	False	True	False	True	False	False	False	False	True	True
3	False	False	True	True	False	True	False	False	False	True	True	True	False	False	False	False
4	True	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False

Product Frequency



Association Analysis

Association analysis is an unsupervised learning tool that looks for hidden patterns and relationships in large datasets. These relationships can take two forms: frequent item sets, or association rules. Frequent item sets are a collection of items that frequently occur together. An itemset is considered as "frequent" if it meets a user-specified support threshold. For instance, if the support threshold is set to 0.4 (40%), a frequent itemset is defined as a set of items that occur together in at least 40% of all transactions

in the database.The second way to view hidden relationships is association rules. Association rules suggest that a strong relationship exists between two items.

Association rules can be thought of as IF-THEN relationships. If a customer purchases item A, the chances of item B being selected by the same customer is determined.

Association rules have two components:

1. Antecedent (IF): An item/group of items that can be found in Itemsets or Datasets.
2. Consequent (THEN): An item that comes with an Antecedent/group of Antecedents.

Association can be measured in the following ways:

1. Support: provides the fraction of transactions that include items A and B. Primarily, Support informs us about frequently purchased items or combinations of items purchased frequently. As a result, we can filter out items with a low frequency.
2. Confidence: indicates how frequently items A and B occur together, given the number of times A occurs. It is the probability of a product being bought (consequent) given that the transaction already has another product (antecedent). It is calculated by dividing the probability of a transaction with both products together by the probability of the antecedent

   That is, Confidence(Milk =>Dill) = Support(Milk,Dill) / Support(Milk).

   If confidence is 1 then both products are always together in all transactions.
3. Lift: Lift indicates the strength of a rule over the random occurrence of A and B.It basically tells us how strong a rule is. The higher the Lift the greater the strength of a rule. Assume for A -> B, the lift value is 4. It means that if you buy A the chances of buying B is 4 times more likely than buying A alone.

   If a rule has a lift of 1, it means that the probability of occurrence of the antecedent and the probability of occurrence of the consequent are independent of each other. When two events are independent of one another, no rule involving those two events can be drawn.

If the lift is > 1, it tells us the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets.

If the lift is < 1, it means the items are substitutes for each other. That is, the presence of one item has a negative effect on the presence of the other item and vice versa.

4. Leverage: measures the difference of A and B appearing together in the data set and what would be expected if A and B were statistically dependent. The rationale in a sales setting is to find out how many more units (items A and B together) are sold than expected from the independent sales.
5. Conviction: compares the probability that A appears without B if they were dependent with the actual frequency of the appearance of A without B.
6. Coverage: measures how often an item set appears in the data set.

Market basket analysis is one application of association analysis.

Market Basket Analysis with Apriori algorithm

The Apriori algorithm was proposed by Agrawal and Srikant in 1994. It is a data mining technique to find associative rules between a combination of items. It is usually used for the analysis of goods purchased with the goal of discovering how likely a customer is to purchase one type of item in association with some other item, so that businesses can develop marketing strategies for interconnected items

Apriori algorithm posits that any subset of a frequent itemset must also be frequent. An itemset that has a support value greater than a threshold value is a frequent itemset It is the algorithm behind Market Basket Analysis.

The steps taken to perform a market basket analysis on our dataset using the Apriori algorithm are documented below:

1. Install Mlxtend: Mlxtend (machine learning extensions) is a Python library that provides useful tools for day-to-day data science tasks. It has an implementation of the Apriori algorithm for extracting frequent item sets for further analysis.

```
pip install mlxtend
```

2. Import libraries

```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

3. Create itemsets of the items that occur in all the transactions:

Note, Support is the ratio of transactions involving the particular product(s) to the total number of transactions

Here, support is set to 10% (0.1) and the length of the itemsets are indicated. This showcases the single, double and triple itemsets.

```
frequent_itemsets = apriori(df, min_support = 0.1,
use_colnames=True)
frequent_itemsets['length'] =
frequent_itemsets['itemsets'].apply(lambda x: len(x))
frequent_itemsets
```

[8]:

	support	itemsets	length
0	0.383383	(Apple)	1
1	0.384384	(Bread)	1
2	0.420420	(Butter)	1
3	0.404404	(Cheese)	1
4	0.407407	(Corn)	1
...	...	...	...
164	0.101101	(chocolate, Nutmeg, Ice cream)	3
165	0.101101	(Onion, chocolate, Ice cream)	3
166	0.100100	(Milk, Kidney Beans, Nutmeg)	3
167	0.101101	(Kidney Beans, Yogurt, Nutmeg)	3
168	0.104104	(Milk, Yogurt, chocolate)	3

169 rows × 3 columns

4. We can then filter significant elements, for example, double itemset elements for which the support is greater than or equal to the 0. 2 threshold.

```
frequent_itemsets[ (frequent_itemsets['length'] == 2) &
                   (frequent_itemsets['support'] >= 0.2) ]
```

[11]:

	support	itemsets	length
49	0.207207	(Ice cream, Butter)	2
50	0.202202	(Kidney Beans, Butter)	2
57	0.202202	(chocolate, Butter)	2
62	0.200200	(Cheese, Kidney Beans)	2
107	0.202202	(chocolate, Ice cream)	2
120	0.211211	(Milk, chocolate)	2

As we can see, the table above returns double item sets with support greater than 20%

5. Create association rules with confidence as a metric. Here, we set confidence to 30%

```
rules1 = association_rules(frequent_itemsets, metric="confidence",
min_threshold=0.30)
```

6. Select top 10 itemsets based on confidence metric

```
rules1 = rules1.sort_values(['confidence'], ascending=False)
rules1[1:11]
```

[26]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
317	(Milk, Dill)	(chocolate)	0.190190	0.421421	0.114114	0.600000	1.423753	0.033964	1.446446
282	(Cheese, Dill)	(Onion)	0.177177	0.403403	0.102102	0.576271	1.428523	0.030628	1.407968
316	(chocolate, Dill)	(Milk)	0.199199	0.405405	0.114114	0.572864	1.413065	0.033358	1.392051
250	(Kidney Beans, Ice cream)	(Butter)	0.196196	0.420420	0.110110	0.561224	1.334913	0.027625	1.320902
313	(Ice cream, Dill)	(chocolate)	0.185185	0.421421	0.103103	0.556757	1.321140	0.025062	1.305330
287	(Cheese, Ice cream)	(Kidney Beans)	0.187187	0.408408	0.104104	0.556150	1.361749	0.027655	1.332863
283	(Cheese, Onion)	(Dill)	0.185185	0.398398	0.102102	0.551351	1.383920	0.028325	1.340919
241	(Sugar, Apple)	(Butter)	0.182182	0.420420	0.100100	0.549451	1.306907	0.023507	1.286384
262	(Butter, Unicorn)	(Ice cream)	0.182182	0.410410	0.100100	0.549451	1.338783	0.025331	1.308601
332	(Milk, Nutmeg)	(Kidney Beans)	0.182182	0.408408	0.100100	0.549451	1.345346	0.025695	1.313045

By this metric, any individual purchasing Milk and Dill will have a 60% probability of also wanting to purchase Chocolate. This is one of the highest probabilities, the pairing is also supported by the 57% probability of the customer getting Milk if they have already purchased Chocolate and Dill.

7. Add the number of items in the antecedents and consequents parts and see the first 5 lines:

```
rules1["antecedent_len"] = rules1["antecedents"].apply(lambda x: len(x))
rules1["consequents_len"] = rules1["consequents"].apply(lambda x: len(x))
rules1[1:6]
```

[15]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	antecedent_len	consequents_len
315	(Milk, Dill)	(chocolate)	0.190190	0.421421	0.114114	0.600000	1.423753	0.033964	1.446446	2	1
284	(Cheese, Dill)	(Onion)	0.177177	0.403403	0.102102	0.576271	1.428523	0.030628	1.407968	2	1
317	(Dill, chocolate)	(Milk)	0.199199	0.405405	0.114114	0.572864	1.413065	0.033358	1.392051	2	1
249	(Kidney Beans, Ice cream)	(Butter)	0.196196	0.420420	0.110110	0.561224	1.334913	0.027625	1.320902	2	1
313	(Dill, Ice cream)	(chocolate)	0.185185	0.421421	0.103103	0.556757	1.321140	0.025062	1.305330	2	1

8. Repeat steps 4 to 6 above for the lift metric

```
rules2 = association_rules(frequent_itemsets, metric="lift",
min_threshold=1)
rules2 = rules2.sort_values(['lift'], ascending=False)
rules2[1:6]
```

[44]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
322	(Cheese, Dill)	(Onion)	0.177177	0.403403	0.102102	0.576271	1.428523	0.030628	1.407968
401	(Unicorn, Dill)	(chocolate)	0.168168	0.421421	0.101101	0.601190	1.426578	0.030231	1.450764
404	(chocolate)	(Unicorn, Dill)	0.421421	0.168168	0.101101	0.239905	1.426578	0.030231	1.094379
391	(chocolate)	(Milk, Dill)	0.421421	0.190190	0.114114	0.270784	1.423753	0.033964	1.110521
390	(Milk, Dill)	(chocolate)	0.190190	0.421421	0.114114	0.600000	1.423753	0.033964	1.446446
389	(chocolate, Dill)	(Milk)	0.199199	0.405405	0.114114	0.572864	1.413065	0.033358	1.392051
392	(Milk)	(chocolate, Dill)	0.405405	0.199199	0.114114	0.281481	1.413065	0.033358	1.114517
326	(Dill)	(Cheese, Onion)	0.398398	0.185185	0.102102	0.256281	1.383920	0.028325	1.095596
323	(Cheese, Onion)	(Dill)	0.185185	0.398398	0.102102	0.551351	1.383920	0.028325	1.340919
403	(Dill)	(chocolate, Unicorn)	0.398398	0.186186	0.101101	0.253769	1.362984	0.026925	1.090565

Our highest 'lift' pairing is Cheese, Dill and Onion at approximately 1.43, this means customers are 1.43 times more likely to purchase Cheese, Dill and Onion than only Cheese and Dill.

```
rules2["antecedent_len"] = rules2["antecedents"].apply(lambda x:
len(x))
rules2["consequents_len"] = rules2["consequents"].apply(lambda x:
len(x))
rules2[1:6]
```

[45]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	antecedent_len	consequents_len
322	(Cheese, Dill)	(Onion)	0.177177	0.403403	0.102102	0.576271	1.428523	0.030628	1.407968	2	1
401	(Unicorn, Dill)	(chocolate)	0.168168	0.421421	0.101101	0.601190	1.426578	0.030231	1.450764	2	1
404	(chocolate)	(Unicorn, Dill)	0.421421	0.168168	0.101101	0.239905	1.426578	0.030231	1.094379	1	2
391	(chocolate)	(Milk, Dill)	0.421421	0.190190	0.114114	0.270784	1.423753	0.033964	1.110521	1	2
390	(Milk, Dill)	(chocolate)	0.190190	0.421421	0.114114	0.600000	1.423753	0.033964	1.446446	2	1

9. Filter generated rule sets:
    a. Filter first 10 records with an Antecedent item length of 1 and a Confidence value greater than 0.20 and a Lift value greater than 1.

```
rules1[(rules1['antecedent_len'] >= 1) &
       (rules1['confidence'] >= 0.20) &
       (rules1['lift'] > 1) ].sort_values(['confidence'],
ascending=False)[1:10]
```

[19]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	antecedent_len	consequents_len
315	(Milk, Dill)	(chocolate)	0.190190	0.421421	0.114114	0.600000	1.423753	0.033964	1.446446	2	1
282	(Cheese, Dill)	(Onion)	0.177177	0.403403	0.102102	0.576271	1.428523	0.030628	1.407968	2	1
317	(Dill, chocolate)	(Milk)	0.199199	0.405405	0.114114	0.572864	1.413065	0.033358	1.392051	2	1
250	(Ice cream, Kidney Beans)	(Butter)	0.196196	0.420420	0.110110	0.561224	1.334913	0.027625	1.320902	2	1
313	(Dill, Ice cream)	(chocolate)	0.185185	0.421421	0.103103	0.556757	1.321140	0.025062	1.305330	2	1
285	(Cheese, Ice cream)	(Kidney Beans)	0.187187	0.408408	0.104104	0.556150	1.361749	0.027655	1.332863	2	1
283	(Cheese, Onion)	(Dill)	0.185185	0.398398	0.102102	0.551351	1.383920	0.028325	1.340919	2	1
263	(Butter, Unicorn)	(Ice cream)	0.182182	0.410410	0.100100	0.549451	1.338783	0.025331	1.308601	2	1
242	(Apple, Sugar)	(Butter)	0.182182	0.420420	0.100100	0.549451	1.306907	0.023507	1.286384	2	1

For the row with ID 282, the probability of seeing the (Cheese, Dill) itemset and Onion together is 10% (0.102102) (support). 58% (0.576271) of people who bought Cheese and Dill also bought Onion (confidence). The sales of Onion in the shopping carts containing Cheese and Dill has increased by 1.428523 (lift). The sales of the (Cheese, Dill) itemset and Onions item is 3% (0.030628) higher than when both items are purchased separately (leverage). The (Cheese, Dil) itemset and Onions are related to each other with a value of 1.407968 (conviction)

b. Filter first 10 records with the Antecedents item name Bread, sorted by Confidence metric

```
rules1[rules1['antecedents'] ==
{'Bread'}].sort_values(['confidence'], ascending=False)[1:10]
```

[19]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	antecedent_len	consequents_len
56	(Bread)	(chocolate)	0.384384	0.421421	0.185185	0.481771	1.143204	0.023197	1.116453	1	1
40	(Bread)	(Ice cream)	0.384384	0.410410	0.181181	0.471354	1.148495	0.023426	1.115283	1	1
30	(Bread)	(Butter)	0.384384	0.420420	0.180180	0.468750	1.114955	0.018577	1.090973	1	1
50	(Bread)	(Sugar)	0.384384	0.409409	0.179179	0.466146	1.138581	0.021809	1.106277	1	1
49	(Bread)	(Onion)	0.384384	0.403403	0.178178	0.463542	1.149077	0.023116	1.112102	1	1
45	(Bread)	(Milk)	0.384384	0.405405	0.174174	0.453125	1.117708	0.018343	1.087259	1	1
35	(Bread)	(Corn)	0.384384	0.407407	0.174174	0.453125	1.112216	0.017573	1.083598	1	1
33	(Bread)	(Cheese)	0.384384	0.404404	0.173173	0.450521	1.114035	0.017726	1.083928	1	1
46	(Bread)	(Nutmeg)	0.384384	0.401401	0.171171	0.445312	1.109394	0.016879	1.079164	1	1

For row 56 above, a customer who purchases bread is 48% (confidence) likely to also purchase chocolate and the probability of seeing bread and chocolate together is approximately 19% (support).

Conclusion

The benefits of having a clear cut understanding of your customer's purchasing needs and behaviors, help organizations make informed decisions that can service their customers effectively.

Apart from its large usefulness in retailing companies, Market Basket Analysis can also be used in several other areas such as the Banking or Forensic sector with credit card purchase data being analyzed for fraudulent patterns, and the Manufacturing industry where it has been put to use for predictive maintenance of machinery.

References

https://www.edureka.co/blog/apriori-algorithm/

https://en.wikipedia.org/wiki/Association_rule_learning

https://michael.hahsler.net/research/recommender/associationrules.html