Analytics for Business LAB

Project Work - Methogological Process

Group 3

Ekaterina Akchurina	10781341
Berk Ceyhan	10761821
Agnieszka Dymska	10758647
Francesco Giurleo	10635473
Anastasiya Harbatovich	10752472
Esteban Nieves	10773742
Aubin Tom Massart	10822089

3 MBA: Expected outcome

The goal of this part is to conduct a market basket analysis (MBA) of the COOP customers' transactions in order to find rules and optimize selling.

Brainstorming phase

Before starting writing a code for MBA, it was essential to aknowledge ourselves with provided datasets. We were given two datasets: **'Products_DB'**, which includes all product IDs with their description and **'Tickets_DB'**, which includes more than 3 million tickets of COOP's customers. Our idea is to create a new dataset in which all the products will be organized in wider groups so that the MBA could be conducted twofolds:

- analyzing raw dataset with products;
- analyzing created groups of products.

Both of these analysis can be essential and give different results.
For instance, we can create a group named 'Water' that comprises sparkling water, lightly sparkling water, naturally sparkling water, and still water.

Products_DB

	Prod_id	Description
144	20101	Sparkling water
145	20102	Lightly sparkling water
146	20103	Naturally sparkling water
147	20104	Still water

Products_DB_groups

	Prod_id	Group
144	20101	Water
145	20102	Water
146	20103	Water
147	20104	Water

Choosing coding language

We chose to code on **Google Colab interface** to be able to collaborate on the same notebook. As our coding language, we decided to proceed with **Python**.
Here are the different tools we used for this analysis:

- Dataset architecture: Pandas, Numpy
- MBA: Apriori library, association_rules library
- Vizualisation : Seaborn, Matplotlib

Later we will dig deeper into used tools and libraries.

Data exploration in Python

First, we displayed the size of a each dataset using df.shape() function:

- Products_DB has **874 rows** and **2 columns** (products_id and description)
- Tickets_DB contains **3807587 rows** and **2 columns** (tickets_id and products_id)

Afterwards, we investigated the existence of Null values in our datasets using **df.isnull().values.any()** function: no null values appeared so our datasets are already cleaned from this point of view.
We also checked column types using **df.info()** to prevent computation issues that could appear during our MBA analysis. IDs are integers and descriptions are strings, so they are in good format.

3 MBA: Steps

Now we would like to present the steps of Market Basket Analysis that we did in order to get the desired results.

Preprocessing

To be able to analyze rules between products/groups, we first need to transform data.
To do so, we **merged group/product and ticket datasets** based on product_id using the **pd.merge()** function. We also added a 'Quantity' column in which we count the appearance of occurences of groups/items on each ticket.
From this dataframe, we could create a new dataset in which each row represents one ticket, each column represents a product/group type. Using an encoding function, each cell takes either the value 1 if the product/group product appears at least once on the ticket, else 0:

```
#merge the datasets and add the column Quantity
df_group= pd.merge(groups, ticket, on='prod_id', how='inner')
df_group.insert(3, "Quantity", 1, True)
df_group
```

```
#encode the different items
basket_groups = (df_group.groupby(['ticket_id', 'Group'])['Quantity']
                 .sum().unstack().reset_index().fillna(0).set_index('ticket_id'))
```

```
#give 1 if the product is present and 0 otherwise and drop rows with just 1 item
def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1
basket_sets_groups = basket_groups.applymap(encode_units)
basket_sets_groups =basket_sets_groups[(basket_sets_groups>0).sum(axis=1)>=2]
```

Description	Acids	Alcohol	Alcohol-free beer	Alcoholic aperitivo drinks	Ammonium	Apples	Apricot	Artichoke	Asparagus	Baby biscuits	...	Women perfume	Yeast	Yogurt dessert	Zucchini	babyfood	chips	kitchenware
ticket_id																		
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
299996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0
299997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0
299998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
299999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
300000	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

Data mining using Apriori

We can now apply Apriori method using apriori Python library. This enables to compute the purchased frenquencies for each group/products based on the ticket dataset.

That is how we used this function in Python before computing MBA rules:

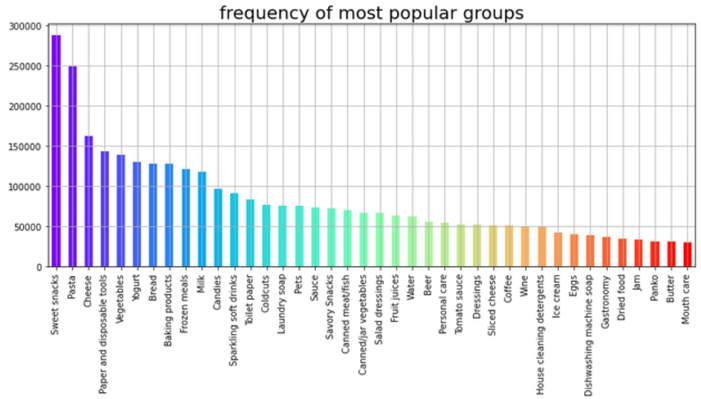
```
frequent_itemsets_groups = apriori(basket_sets_groups, min_support=0.0005, max_len=2,
                                   use_colnames=True).sort_values('support', ascending=False).reset_index(drop=True)

frequent_itemsets_groups['lenght']=frequent_itemsets_groups['itemsets'].apply(lambda x:len(x))
frequent_itemsets_groups
```

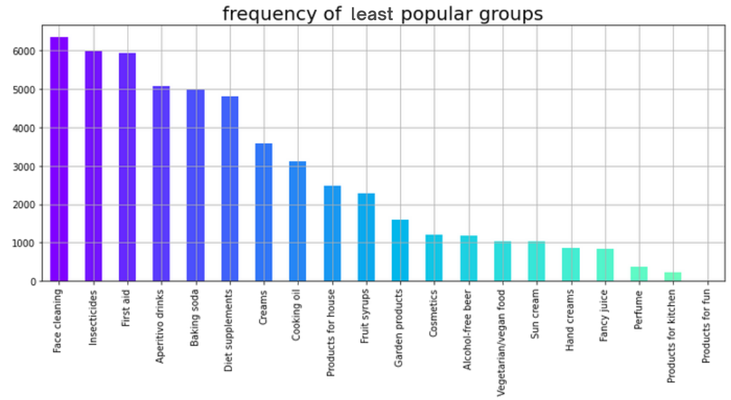
3 MBA: Steps

Presentation of the steps of Market Basket Analysis that we did in order to get the desired results.

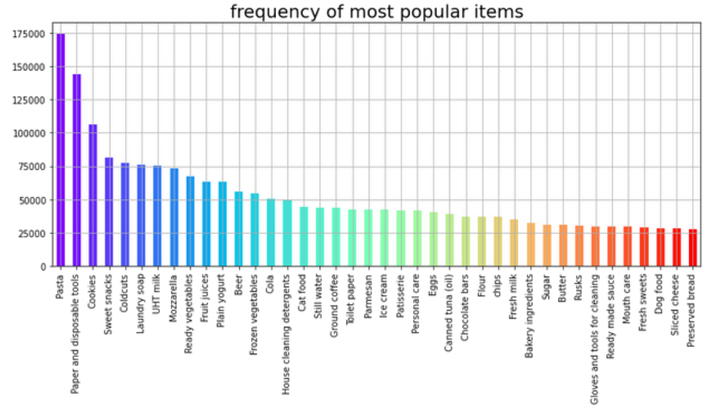
Analysis of frequency

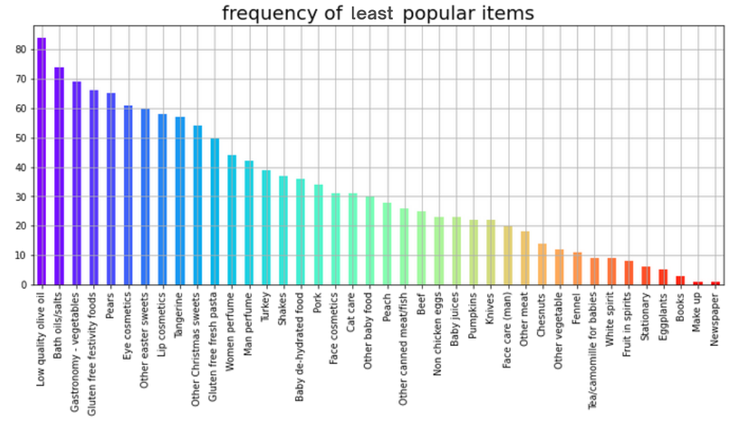
We checked the **most and least frequently bought group products** to better understand the purchasing behaviour of customers:





We also checked those frequencies for non-grouped products (items):





We could see that these results are different for product groups and items, so splitting our analysis in these two parts seems relevant.

3 MBA: Steps

Presentation of the steps of Market Basket Analysis that we did in order to get the desired results.

Analysis of main rules

Then, we created rules to achieve marketing insights. The goal was to get the most interesting rules between items/groups regarding **lift**, **confidence** & **support.**
We firstly computed the rules for high lift, high supports and confidence higher than 30% ordered by lift descending:

```
#groups sorted by lift with only high support of antecedents
rules_groups = association_rules(frequent_itemsets_groups, metric="lift",
                                 min_threshold = 1).sort_values('lift',ascending=False).reset_index(drop=True)
filtered_rules_groups = (rules_groups[(rules_groups['antecedent support'] > 0.05) &
                        (rules_groups['consequent support'] > 0.05) &
                        (rules_groups['confidence'] > 0.3)]).sort_values('lift',ascending=False).reset_index(drop=True)
```

Here is the output of this function:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Shampoo and hair treatments)	(Personal care)	0.073087	0.136110	0.030179	0.412917	3.033687	0.020231	1.471494
1	(Mouth care)	(Personal care)	0.086384	0.136110	0.030084	0.348255	2.558624	0.018326	1.325503
2	(Gloves and tools for cleaning)	(House cleaning detergents)	0.080829	0.127960	0.025156	0.311222	2.432174	0.014813	1.266067
3	(House cleaning detergents)	(Laundry soap)	0.127960	0.185963	0.053413	0.417422	2.244645	0.029617	1.397300
4	(Dishwashing machine soap)	(Laundry soap)	0.120758	0.185963	0.046797	0.387523	2.083868	0.024340	1.329090
...	...	...	...	...	...	...	...	...	...

Based on such results in managerial report we proposed some ideas for marketing strategies to increase sales.
The same analysis we did for products with lower support of antecedent products, in this way we aimed at increasing sales of products that are not frequently purchased at COOP.

Analysis of specific groups and items

As another step of our analysis we decided to search for specific groups of products and try to find some interesting insights. With this we tried to increase sales of these products by proposing marketing strategies.

We computed rules with a minimum confidence of 0.1, ordered by lift descending. As shown in the below picture we decided to search for garden products.

```
#sorted by garden products
rules_groups = association_rules(frequent_itemsets_groups, metric="lift",
                                 min_threshold = 0).sort_values('lift',ascending=False).reset_index(drop=True)
filtered_rules_groups = (rules_groups[(rules_groups['antecedent support'] > 0.00) &
                        (rules_groups['consequent support'] > 0.00) &
                         (rules_groups['antecedents'] == frozenset({'Garden products'}))  )&
                        (rules_groups['confidence'] > 0.1)]).sort_values('lift',
                                                ascending=False).reset_index(drop=True).head(20)
```

Here are the top 5 results we obtained with above code.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Garden products)	(Beer)	0.005274	0.155358	0.001322	0.250668	1.613490	0.000503	1.127194
1	(Garden products)	(Wine)	0.005274	0.134545	0.001075	0.203877	1.515304	0.000366	1.087087
2	(Garden products)	(Pets)	0.005274	0.121354	0.000941	0.178476	1.470704	0.000301	1.069532
3	(Garden products)	(Gloves and tools for cleaning)	0.005274	0.080829	0.000599	0.113636	1.405891	0.000173	1.037014
4	(Garden products)	(Savory Snacks)	0.005274	0.175839	0.001206	0.228610	1.300106	0.000278	1.068410

We also investigated other products, like sun creams, baby products and vegetarian/vegan products. We assumed that garden products and sun creams fall into category of seasonal products whilst baby products and vegetarian/vegan products can be seen as products for specific groups of customers. We also proposed marketing strategies to increase their sales.

3 MBA: Steps

Presentation of the steps of Market Basket Analysis that we did in order to get the desired results.

Searching for item combinations

As a next step, we investigated the raw dataset given, so without groups, in order to find rules related to items.
First, we looked at pairs of items with high lifts and high confidence:

```
#high support and high confidence specific products
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold = 0
                          ).sort_values('confidence',ascending=False).reset_index(drop=True)
filtered_rules = (rules[(rules['antecedent support'] > 0.02) &
                        (rules['consequent support'] > 0.02) &
                        (rules['lift'] > 1)]).sort_values('confidence',ascending=False
                                                          ).reset_index(drop=True).head(20)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Mascarpone)	(Cookies)	0.020804	0.274742	0.011684	0.561623	2.044186	0.005968	1.654417
1	(Toilet paper)	(Paper and disposable tools)	0.146343	0.305454	0.078820	0.538598	1.763273	0.034119	1.505296
2	(Peeled tomato)	(Pasta)	0.028739	0.299112	0.014969	0.520866	1.741375	0.006373	1.462823
3	(Dishwashing soap)	(Paper and disposable tools)	0.087300	0.305454	0.043585	0.499255	1.634469	0.016919	1.387025

As we can see, the top rules we found were mostly related to pair of items bought to cook italian dishes. Therefore, we decided to dig deeper into the combinations of items related to the most famous italian recipes as we know that they are very important in the country.

Searching for Italian recipes

Using the same query by filtering antecedent by Mascarpone:

```
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold = 0
                          ).sort_values('confidence',ascending=False).reset_index(drop=True)
filtered_rules = (rules[(rules['antecedent support'] > 0.000) &
                        (rules['consequent support'] > 0.00) &
                        (rules['antecedents'] == frozenset({'Mascarpone'})  )&
                        (rules['lift'] > 1)]).sort_values('confidence',ascending=False
                                                          ).reset_index(drop=True).head(10)
```

We discovered that the top 1 rule in terms of high lift and confidence is Mascarpone and Cookies, that are both used to cook Tiramisu:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Mascarpone)	(Cookies)	0.020804	0.274742	0.011684	0.561623	2.044186	0.005968	1.654417

We also got the same results for example on White rice - Broth preparation items used to cook Risotto, Canned tuna (oil) - Pasta for Pasta al tonno, and others.

Then, we decided to suggest Coop to develop some marketing actions like for instance suggestion posters to improve sellings regarding those items.