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

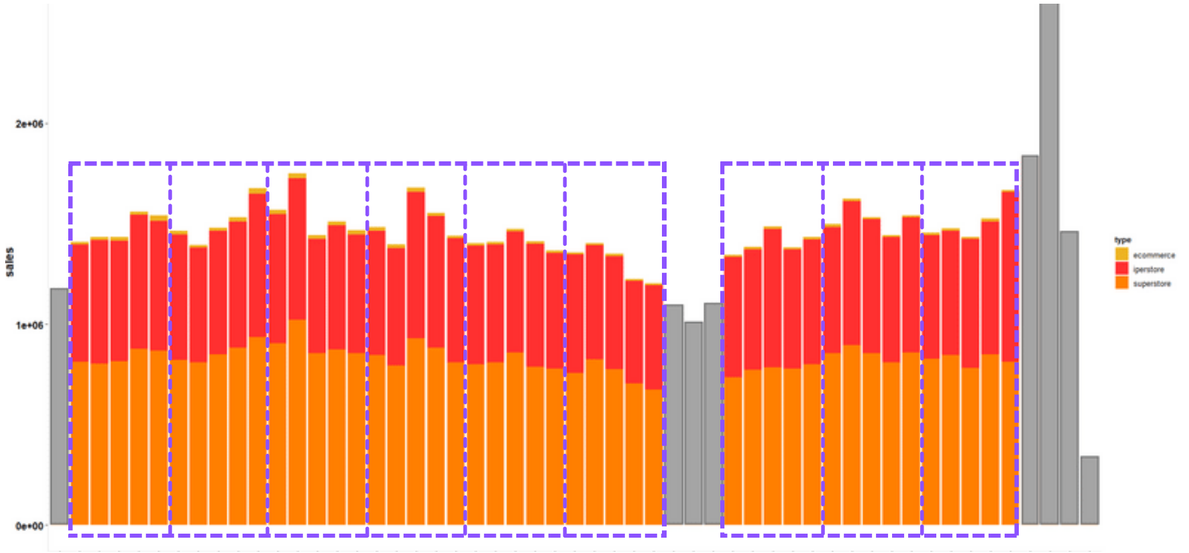
Agenda

1. Customers' Behaviour Analysis
2. Churn Analysis
3. Market Basket Analysis

1 Customers' Behaviour Analysis: Preparation

Time-frame Definition

First of all, it was decided to exclude some weeks (the grey bars in the graph on the right) as they were periods of Christmas and summer holidays.
After that, the total amount of weeks was splitted into 9 periods (5 weeks per period).
The reason was to have 9 periods in order to analyze how customers were behaving **between** periods, and 5 weeks per period in order to have a significant amount of transactions to understand how they were behaving **within** each period.



Anomaly Detection

It was decided to exclude from further analysis 2 typologies of customers:

- those who enrolled in the Coop fidelity program when older than 90 years old since it was assumed to be a weird behavior given this old age;
- those with more than 40.000€ spent in one year.

1 Customers' Behaviour Analysis: RegFM

Define RegFM rules

It was decided to cluster customers according to their Monetary, Frequency, and Regularity values for each of the 9 periods analyzed. For the first 2 metrics, it was decided to assign them a score looking at the percentiles distributions, in fact, the score "Low" correspond to values below the 35th percentile, "Medium" to values between the 35th and the 75th percentiles, "High" to values above the 75th percentile. For what regards Regularity it was decided to assign the scores through this formula:

$$Regularity\ [*] = \frac{n°\ days\ in\ the\ period}{n°\ transactions} - n°\ days\ since\ the\ last\ purchase$$

	Low	Medium	High
Monetary [€]	<=165	>165 & <=455	>455
Frequency [transactions]	<=5	>5 & <=12	>12
Regularity [*]	<=-5	>-5 & <=5	>5

Customers Clustering

	Monetary	Frequency	Regularity
Champions	High	Medium-High	Medium-High
Loyal Customer	Medium	Medium-High	Medium-High
Wholesalers	High	Low	Medium-High
Promising	Low	Medium-High	Medium-High
Can't Lose Them	Medium-High	Medium-High	Low
Need Attention	Medium-High	Low	Low
Recent Users	Medium	Low	Medium-High
New Customers	Low	Low	High
Partial Churner	Low	Medium-Low	Medium-Low

After having defined some rules to assign scores to the Monetary, Frequency, and Regularity metrics, it was decided to create clusters of customers with similar characteristics. The result was a total amount of 9 customer clusters, from Champions (the best) to Partial Churner (the worst). In addition, the 10th cluster, which here is not reported, is Total Churner, which means a customer with no purchases in one of the nine periods defined, so no values for Monetary, Frequency, and Regularity metrics.

1 Customers' Behaviour Analysis: Longitudinal Analysis - Part 1

Customers Path

After the customer clustering, the results were merged and was created this table. It is possible to identify, for each customer, the cluster to which it belongs from *Period_1* to *Period_9*.

Customer_id	Period_1	Period_2	Period_3	Period_4	Period_5	Period_6	Period_7	Period_8	Period_9
1	New_Customers	New_Customers	New_Customers	Loyal_Customers	Partial_Churner	New_Customers	Partial_Churner	Partial_Churner	Partial_Churner
2	New_Customers	Partial_Churner	New_Customers	Loyal_Customers	Partial_Churner	New_Customers	Promising	Partial_Churner	Can't_Lose_Them
4	Loyal_Customers	Loyal_Customers	Loyal_Customers	Loyal_Customers	Loyal_Customers	Loyal_Customers	Can't_Lose_Them	Loyal_Customers	Champions
5	Champions	Can't_Lose_Them	Can't_Lose_Them	Loyal_Customers	Champions	Loyal_Customers	Loyal_Customers	Champions	Champions

Markov Chain

The customer path table was exploited to perform a Markov Chain analysis, in order to understand transition probabilities from t to t+1 in the different categories analyzed.

From/To	Champions	Loyal_Customers	Whoolesalers	Can.t_Lose_Them	Need_Attention	Promising	Recent_Users	New_Customers	Partial_Churner	Total_Churner
Champions	0.697	0.171	0.039	0.019	0.008	0.005	0.044	0.007	0.008	0.002
Loyal_Customers	0.135	0.599	0.003	0.017	0.006	0.088	0.062	0.036	0.049	0.004
Whoolesalers	0.276	0.029	0.374	0.006	0.028	0.001	0.226	0.029	0.013	0.018
Can.t_Lose_Them	0.248	0.355	0.017	0.021	0.010	0.061	0.084	0.093	0.066	0.045
Need_Attention	0.119	0.137	0.066	0.014	0.035	0.014	0.261	0.166	0.104	0.083
Promising	0.012	0.268	0.001	0.008	0.002	0.443	0.013	0.104	0.136	0.013
Recent_Users	0.087	0.149	0.062	0.010	0.026	0.011	0.374	0.146	0.091	0.045
New_Customers	0.014	0.072	0.005	0.004	0.010	0.059	0.107	0.351	0.218	0.159
Partial_Churner	0.014	0.112	0.005	0.007	0.010	0.099	0.095	0.309	0.242	0.109
Total_Churner	0.009	0.021	0.007	0.002	0.007	0.011	0.062	0.292	0.107	0.484

1 Customers' Behaviour Analysis: Longitudinal Analysis - Part 2

CUSTOMERS CHARACTERISTICS

With a single transition matrix performed on the whole dataset, the result was too general. For this reason, Multiple Markov Chains have been run, filtering through several customers' **characteristics** and different **attributes** per each characteristic:

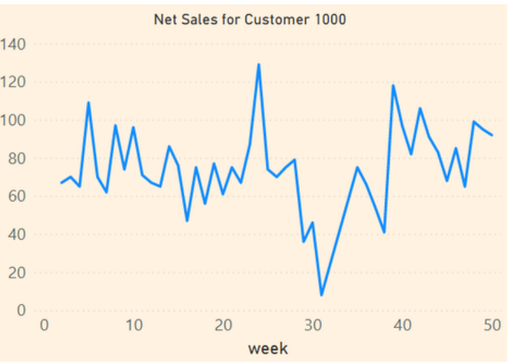
- ***Gender** [Female, Male];*
- ***Age** [Young, Adult, Senior, Old];*
- ***Duration of the Relationship** [Recent, Medium, Long, Very Long];*
- ***Prone to Promotions** [No Prone, Prone, Very Prone].*

2 Churn Analysis: Churn Definition

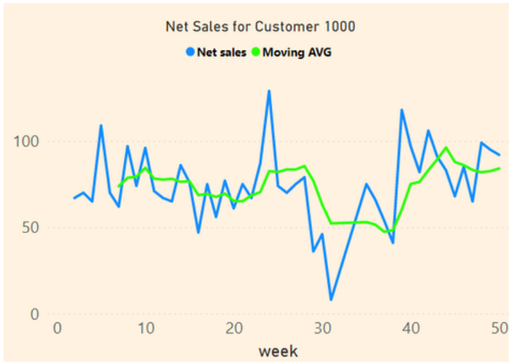
Customer acquisition is generally more costly than customer retention in the retail sector. Thus, companies should try to identify customers with a high risk of churn and run targeted marketing campaigns in order to prevent churn. It is hard to define churn in retail because it is a non-contractual business. Each customer has various purchase behaviors. So, churn should be defined separately for each customer. A churner can be a customer who left the company. Yet sometimes, the customer would not churn immediately. It would be a case of partially reducing the spending before finally moving the whole basket to a competitor store. So, we defined churn in two ways.

- **Soft Churner:** A soft churner is defined as a customer who decreased his/her purchases dramatically.
- **Hard Churner:** A customer who stopped purchasing for a long time is a hard-churner.

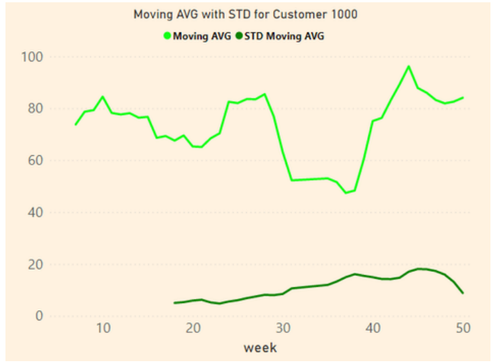
The steps we took are explained in the following graphs. As an example, only the customer id=1000 is shown. These calculations are done separately for each customer.



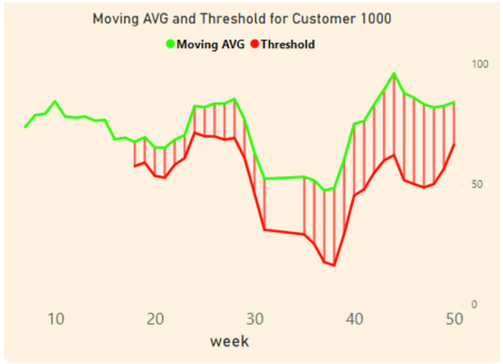
1 Aggregate the customer net sales on a weekly domain



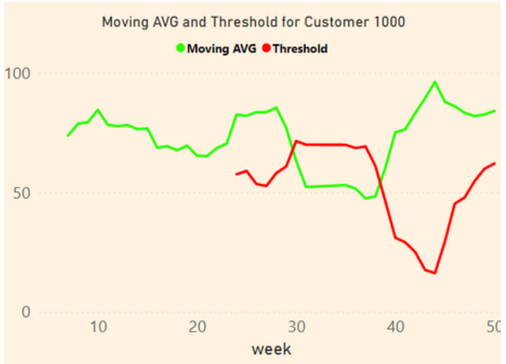
2 Calculate the 6-week moving average of weekly net sales to smoothen out the curve



3 Calculate the 12-week standard deviation of the moving average



4 Define a threshold to the MovAvg
Threshold = MovAvg - 2*StdDev

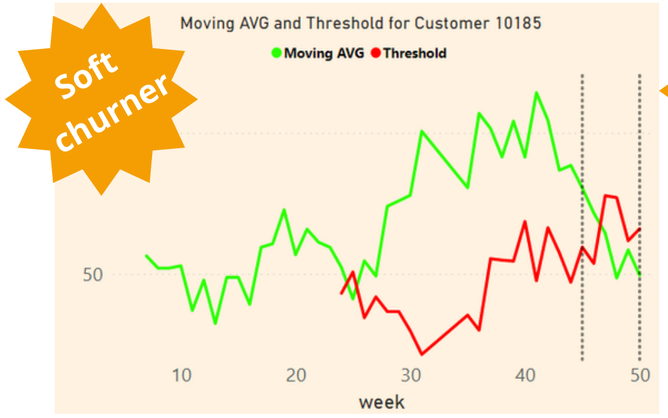


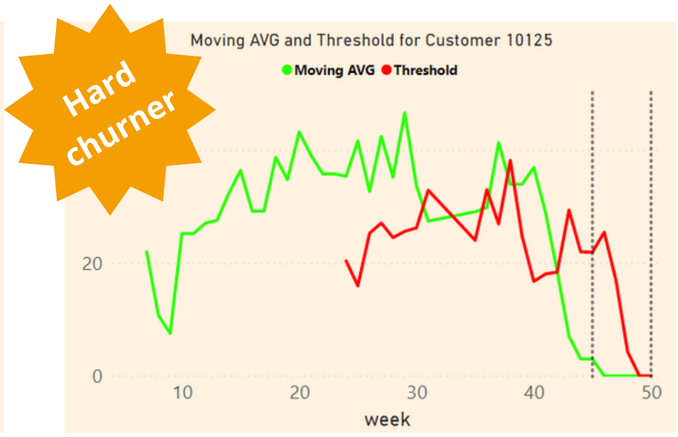
5 Shift the threshold 6 weeks forward as a proactive action

Note 1: The number of weeks in each step is decided by trial and error considering a trade-off between generalization in case of choosing a high number of weeks and high deviation in case of choosing a low number.
Note 2: The definition of threshold corresponds to the 95% confidence interval of the moving average of net sales. Purchases lower than the lower boundary of this region are accepted as a dramatical decrease.

2 Churn Analysis: Churn Definition

Now that we can compare the current purchases with the previous purchases, it is time to define both types of churn analytically. First of all, a selection of a time period to observe is necessary. We chose the last 6 weeks as an observed time period. This way we can identify the churners that we have now as hard churners and we can predict the churners of the following time periods as soft churners.
Again, as a trade-off between too much generalization and high variance, we put a lower threshold on the number of observations of a state in the observed time frame. A soft churner is identified only if the net sales are under the threshold for equal or more than half of the last 6 weeks (so, at least 3 times). The same threshold applies also for the hard churners. The following graphs show examples of these.





$$\text{Soft Churn Percentage} = \frac{\#\text{ Soft Churners}}{\#\text{ All Customers}} = \frac{2396}{25000} = 9.6\%$$

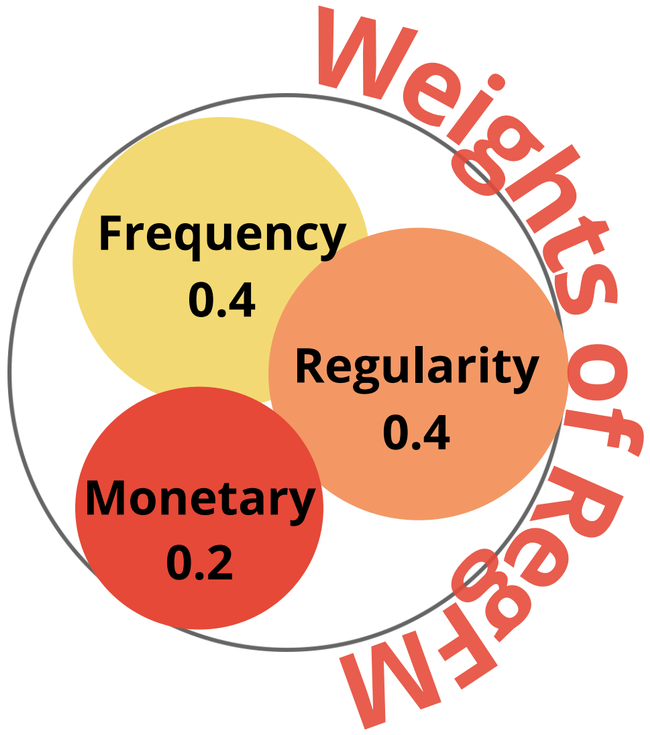
$$\text{Hard Churn Percentage} = \frac{\#\text{ Hard Churners}}{\#\text{ All Customers}} = \frac{415}{25000} = 1.7\%$$

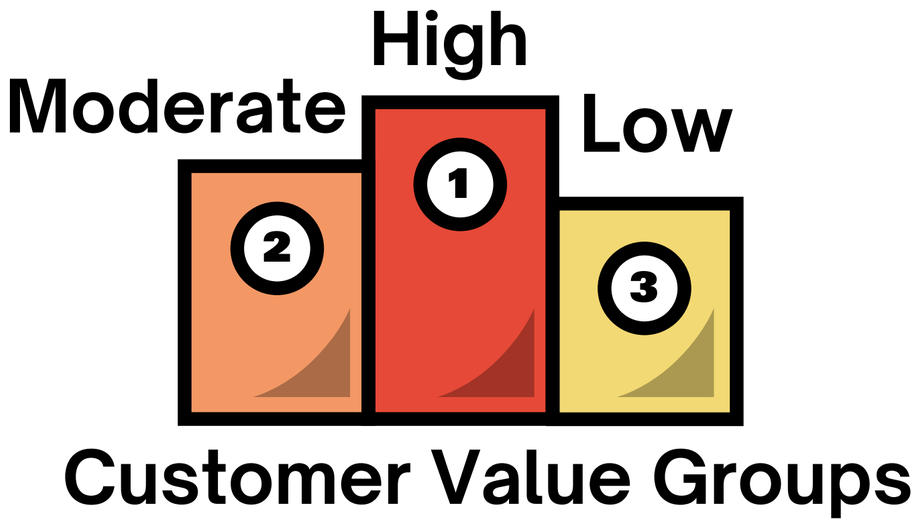
6 Define Soft Churn & Hard Churn in a given period (last 6 weeks)

Soft Churn: Moving average goes below the threshold for more than half of the observed period
Hard Churn: Moving average = 0 for more than half of the observed period

2 Churn Analysis: Customer Value

Then, in order to allocate the marketing budget in a proper way, we created a Customer Value score for every customer to evaluate their importance for Coop. It was calculated as a weighted sum of RegFM scores for each of 9 period (in order to track customers` behaviour changes during the year), and then, their average was taken as a result. Different weights of RegFM parameters were taken because we considered Frequency and Regularity more important factors to charactarize customer loyalty.
Eventually, customer values were divided in 3 categories: High, Medium and Low.





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:

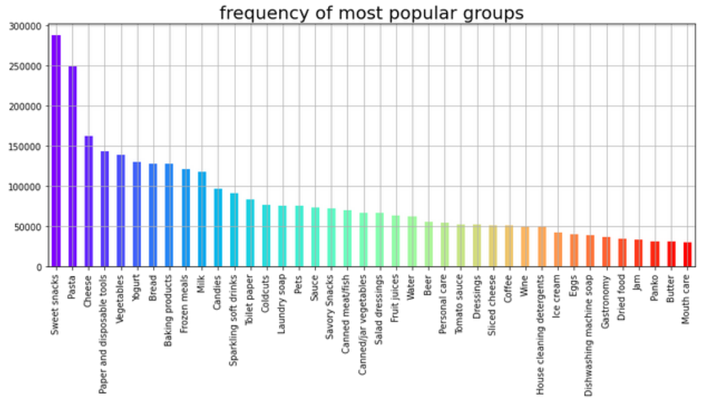
```
frequent_itemsets_groups = apriori(basket_sets_groups, min_support=0.0005, max_len=2,
                                   use_colnames=True).sort_values('support', ascending=False).reset_index(drop=True)

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

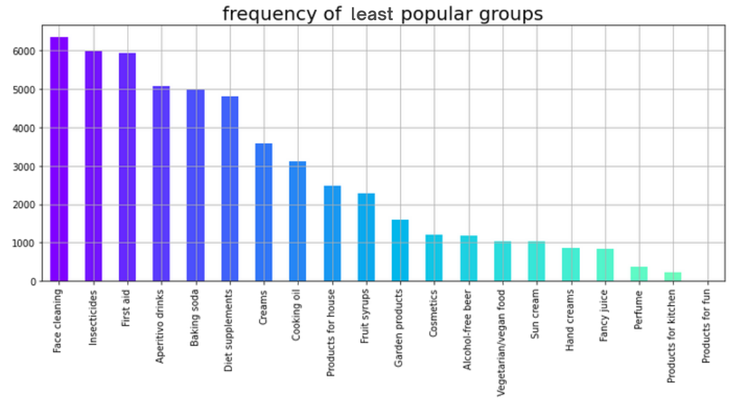
3 MBA: Steps

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

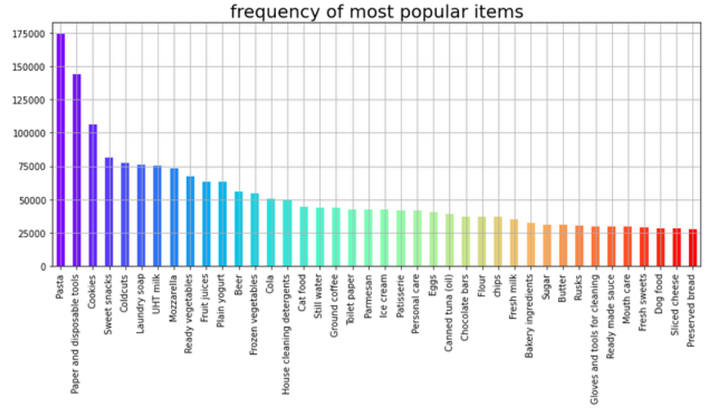
Analysis of frequency

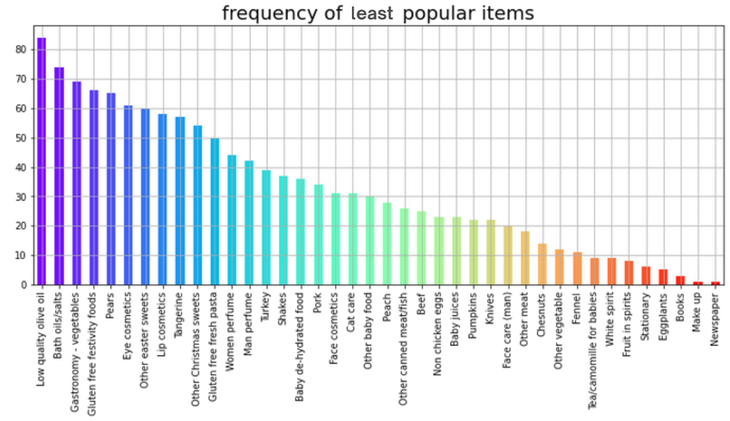
We checked the **most and least frequently bought group products** to better understand the purchasing behaviour of customers:





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.