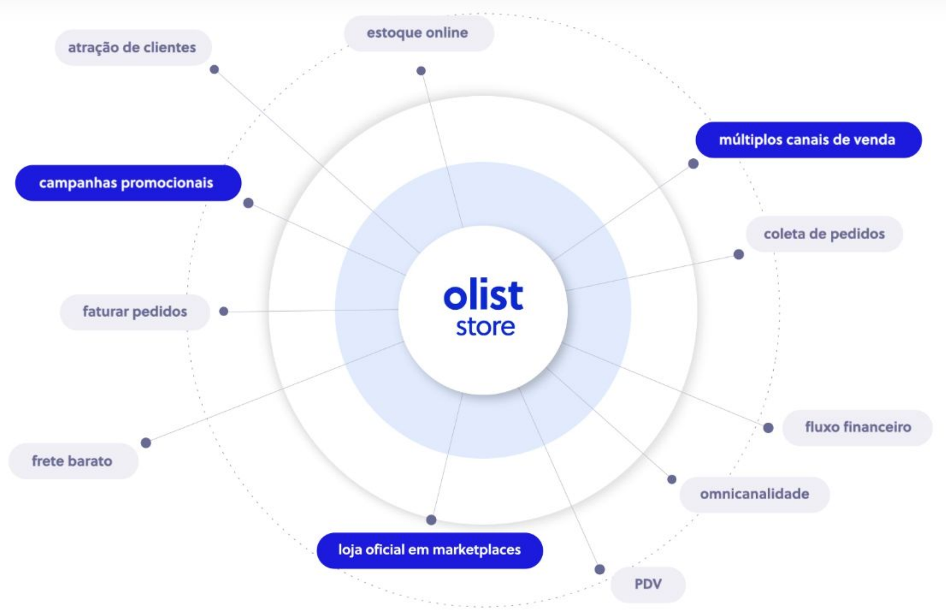
O-list Customer segmentation

Julien LE BOUCHER 05 - 15 - 2023

Context and goals



O-list is a brazilian e-commerce website that provides services to ensure a reliable connection between customers and sellers.

My missions :

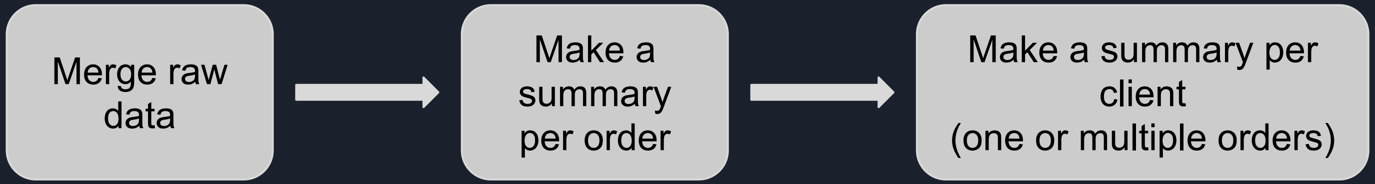
1. Provide a client segmentation to help the marketing team in designing promotional campaigns.

2. Evaluate the need of updating groups through time.

Plan

1. Client information generation.
2. Clients exploratory analysis.
3. Methodology and results of some unsupervised client segmentations.
4. Maintenance.

1 - Generate client information



1 - Generate client information

Merge raw data → Make a summary per order → Make a summary per client (one or multiple orders)

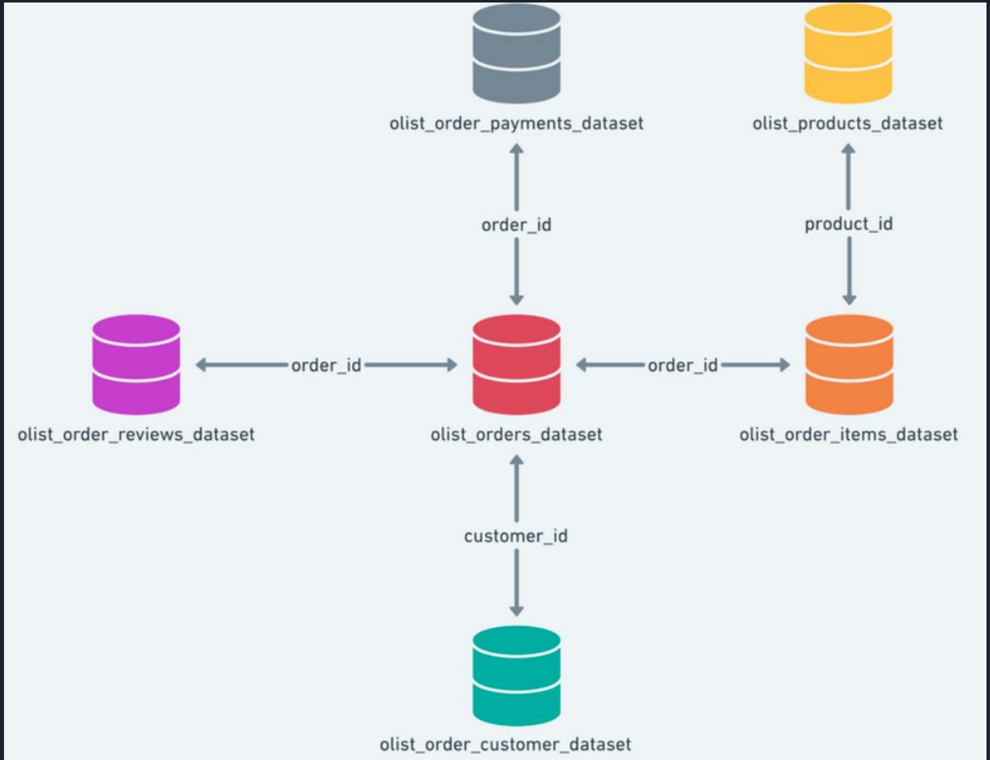
Information spread across 9 datasets

dataset_name	columns_names	rows_num	cols_num	total_duplicates
order_reviews	['review_id', 'order_id', 'review_score', 'review_comment_title', 'review_comment_message', 'review_creation_date', 'review_answer_timestamp']	99224	7	0
order_items	['order_id', 'order_item_id', 'product_id', 'seller_id', 'shipping_limit_date', 'price', 'freight_value']	112650	7	0
sellers	['seller_id', 'seller_zip_code_prefix', 'seller_city', 'seller_state']	3095	4	0
geolocation	['geolocation_zip_code_prefix', 'geolocation_lat', 'geolocation_lng', 'geolocation_city', 'geolocation_state']	1000163	5	261831
orders	['order_id', 'customer_id', 'order_status', 'order_purchase_timestamp', 'order_approved_at', 'order_delivered_carrier_date', 'order_delivered_customer_date', 'order_estimated_delivery_date']	99441	8	0
products	['product_id', 'product_category_name', 'product_name_lenght', 'product_description_lenght', 'product_photos_qty', 'product_weight_g', 'product_length_cm', 'product_height_cm', 'product_width_cm']	32951	9	0
order_payments	['order_id', 'payment_sequential', 'payment_type', 'payment_installments', 'payment_value']	103886	5	0
customers	['customer_id', 'customer_unique_id', 'customer_zip_code_prefix', 'customer_city', 'customer_state']	99441	5	0
product_category_name_translation	['product_category_name', 'product_category_name_english']	71	2	0

Information pruning

dataset_name	columns_names	rows_num	cols_num	total_duplicates
order_reviews	['review_id', 'order_id', 'review_score', ~~'review_comment_title', 'review_comment_message', 'review_creation_date', 'review_answer_timestamp'~~]	99224	7	0
order_items	['order_id', 'order_item_id', 'product_id', 'seller_id', 'shipping_limit_date', 'price', 'freight_value']	112650	7	0
~~[illegible]~~	~~['seller_id', 'seller_zip_code_prefix', 'seller_city', 'seller_state']~~	[illegible]	[illegible]	[illegible]
~~[illegible]~~	~~[illegible] 'geolocation_lng', 'geolocation_city', 'geolocation_state']~~	[illegible]	[illegible]	[illegible]
orders	['order_id', 'customer_id', 'order_status', 'order_purchase_timestamp', 'order_approved_at', 'order_delivered_carrier_date', 'order_delivered_customer_date', 'order_estimated_delivery_date']	99441	8	0
products	['product_id', 'product_category_name', ~~'product_name_lenght', 'product_description_lenght', 'product_photos_qty', 'product_weight_g', 'product_length_cm', 'product_height_cm', 'product_width_cm'~~]	32951	9	0
order_payments	['order_id', 'payment_sequential', 'payment_type', 'payment_installments', 'payment_value']	103886	5	0
customers	['customer_id', 'customer_unique_id', 'customer_zip_code_prefix', 'customer_city', 'customer_state']	99441	5	0
product_category_name_translation	['product_category_name', 'product_category_name_english']	71	2	0

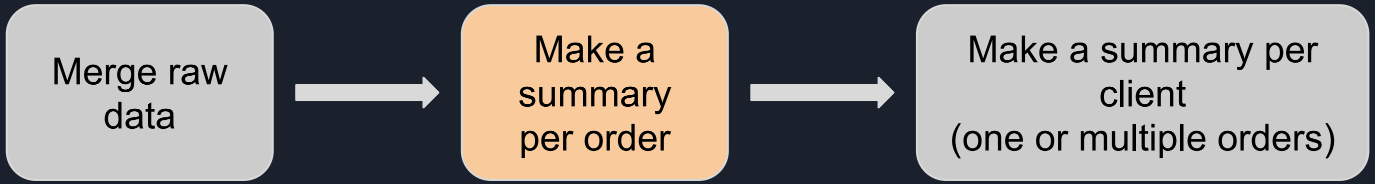
Merged into one dataset



merging schema

- Result = (119 143 rows , 28 columns)
- 99 441 orders.
- dates span : 4 sept of 2016 – 17 oct of 2018 (25 months)
- I made a first exploration to understand the 28 features, and I design functions in order to generate the summary of each order :
  - easy case : one row holds all information.
  - complex cases : understand the meaning of each line in different scenarios.
- Some oddities were found in the data.
  (can be exposed at the end of this presentation.)

1 - Generate client information



37 features in an order summary

Order status, delay for delivery, cost, satisfaction, recency...

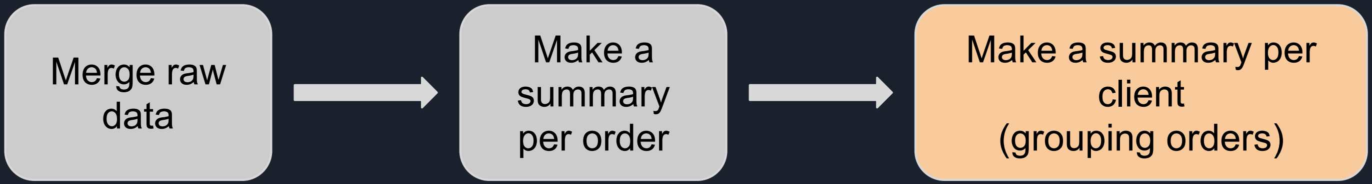
Try and find client's purchasing behavior : moments of activity in the week or the day.

reflect the client's categories of interest.

reflect the client's payment preferences
(+ feature 9)

```
0   order_id
1   order_status
2   binary_order_status
3   purchase_time
4   delivery_time
5   n_items
6   order_cost
7   order_cost_minus_payment
8   review_score
9   payment_installments
10  days_between_purchase_and_delivery
11  hour_of_purchase
12  weekday_of_purchase
13  value_home
14  value_sports_leisure
15  value_electronics_and_multimedia
16  value_unknown
17  value_toys
18  value_auto
19  value_tools_and_professional_material
20  value_health_and_beauty
21  value_pet_shop
22  value_baby
23  value_watches_gifts
24  value_art_cinema_music
25  value_stationery
26  value_fashion
27  value_other
28  value_books
29  value_security
30  freight
31  freight_value
32  payment_value_credit_card
33  payment_value_debit_card
34  payment_value_voucher
35  payment_value_boleto
36  payment_value_not_defined
```

1 - Generate client information



72 features in a client summary.

Important features used in the following clustering models :

- Total value spent by the client (~ Monetary value in RFM).
- Total number of purchases (~ Frequency in RFM).
- days since last purchase (~ Recency in RFM).
- mean of the review scores.
- mean days of delivery per order.
- values and ratios spent by categories.
- values and ratios paid by a certain payment type.
- a dynamic ratio :
  - value spent in the second half being a client / value spent in the first half.
  - 1 if the client is new (because 2 halves does not make sense).

RFM segmentation is explained in the appendices

Some binary features :

- `paid_less_than_due`
- `has_had_a_non_delivered_order`
- `has_contracted_payment_installments`

Purchasing preferences (for multiple time buyers):

- `preferred_week_moment_to_purchase`
- `preferred_day_moment_to_purchase`

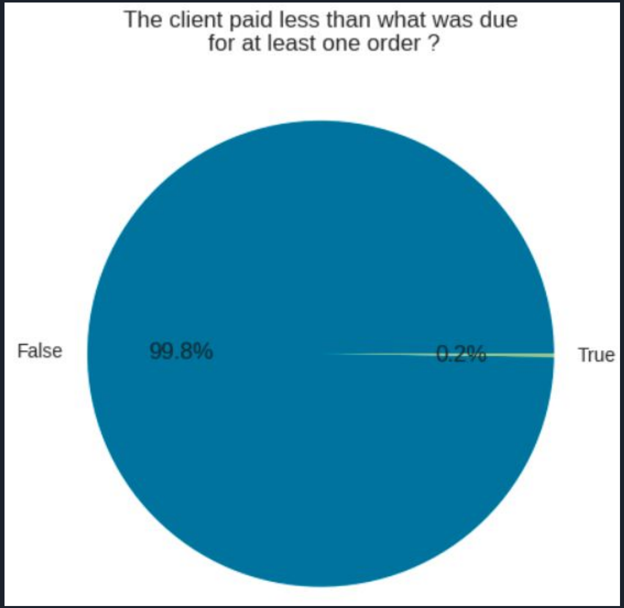
2 - Clients exploratory analysis

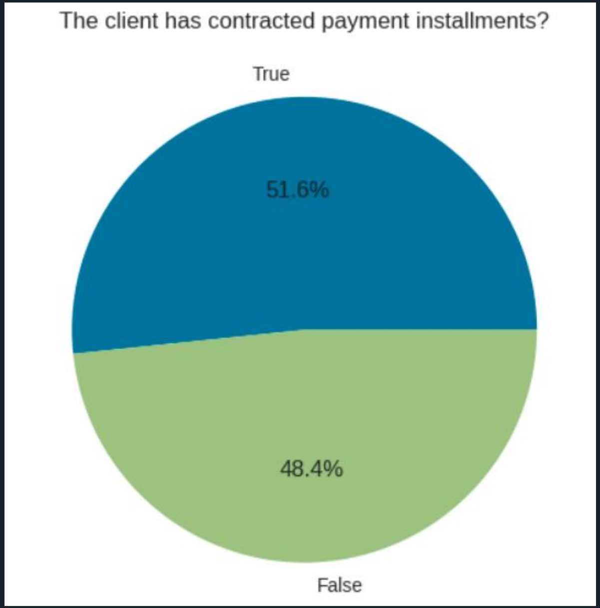
(n_clients = 96 095)

Features unused as input for clustering models.

Though, can be insightful for the marketing team if used as filters for conditional segmentation.

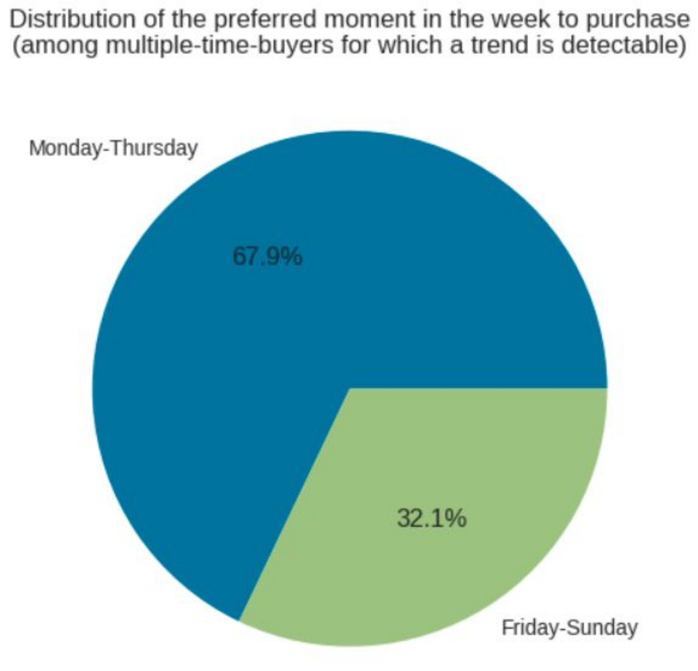
Analysis of the binary features

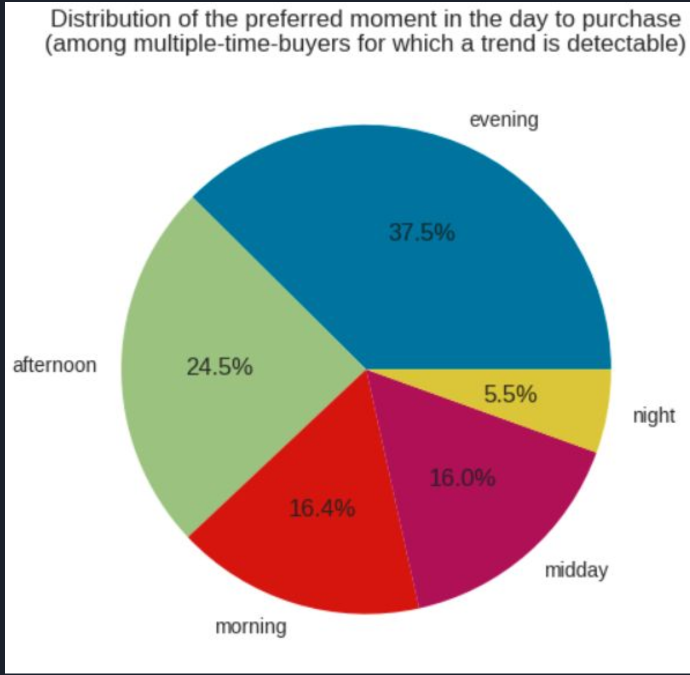




Could be used to make specific client after-sales care.
Or propose a subscription for less fees when paying with installments.

Analysis of the purchasing preferences of the multiple-time buyers (3 % of the clients)

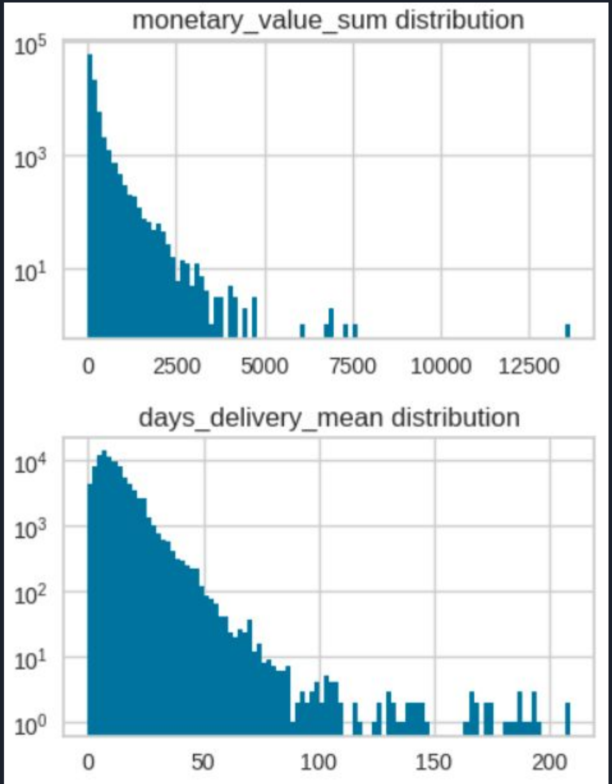




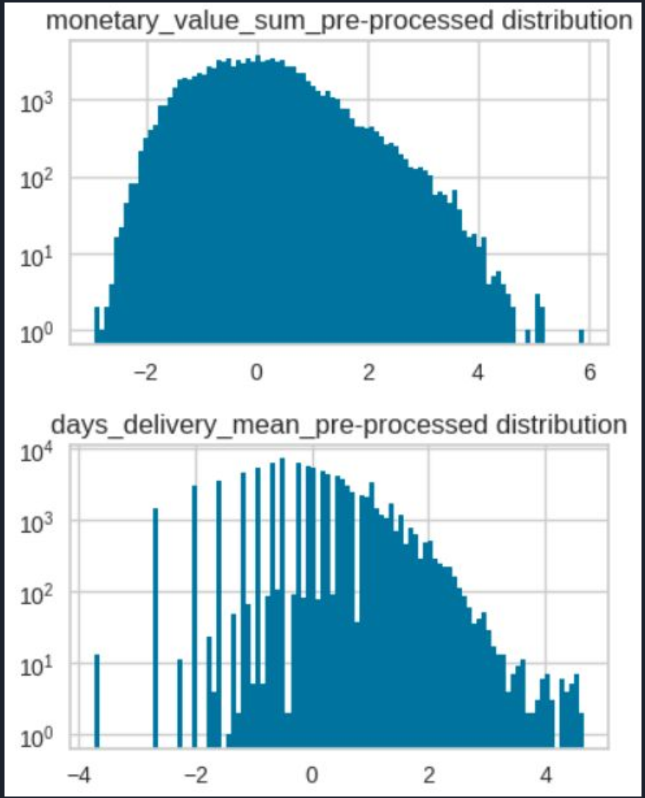
→ Could be used to make offers at the right time to trigger orders.

The 13 Features used as input for segmentation via unsupervised models.

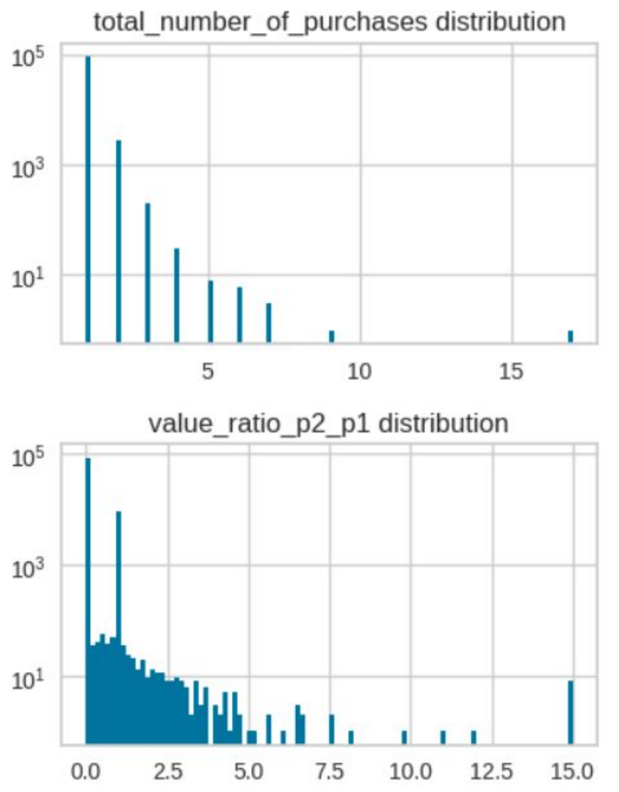
'Total value spent on the platform'
and
'mean days before delivery'



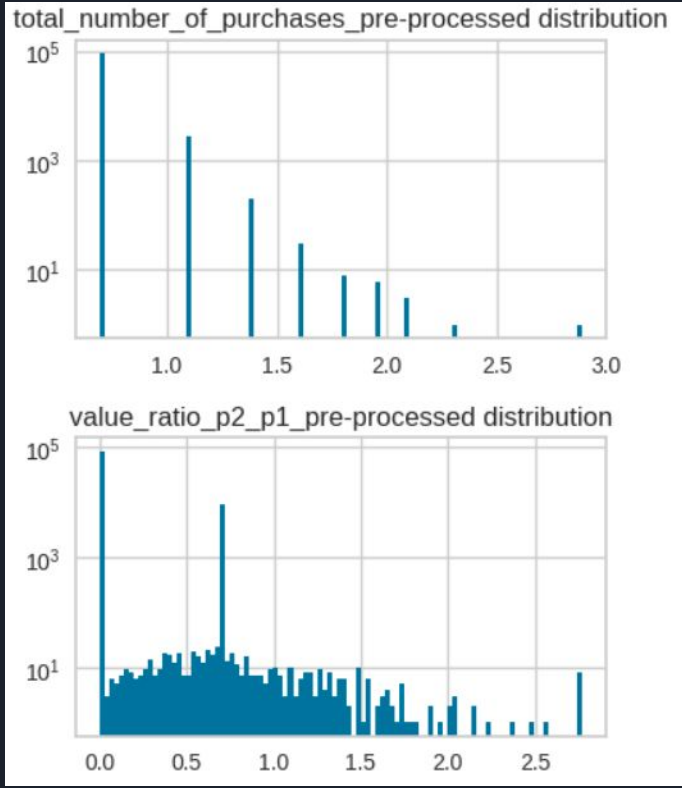
log transformation
+
standardization



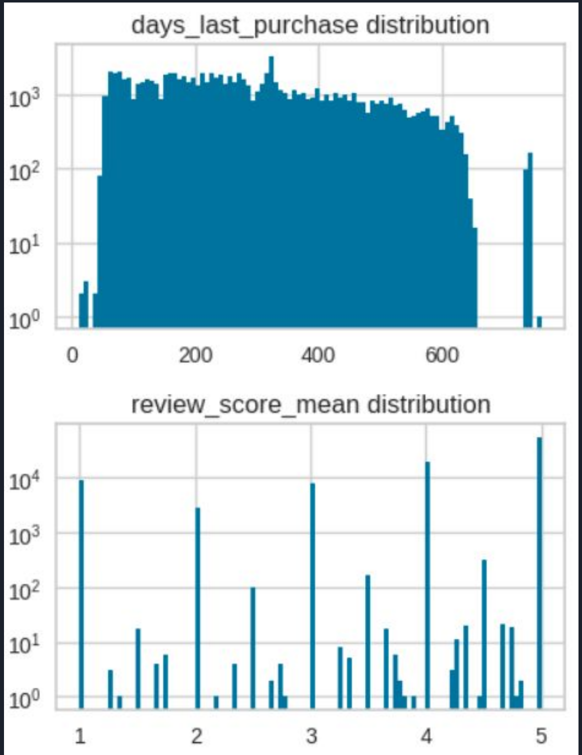
'Total number of purchases'
and
'Dynamic ratio of the client between first and second half of platform fidelity'



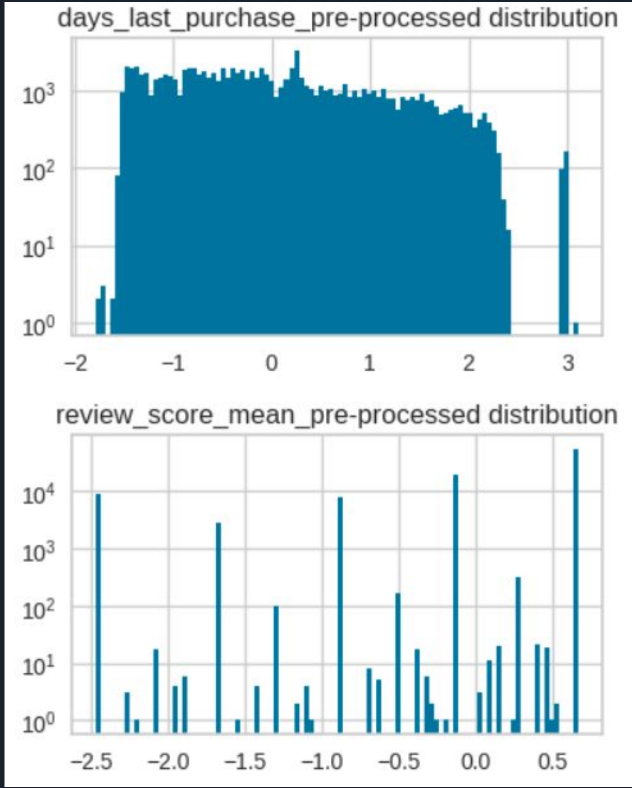
log transformation



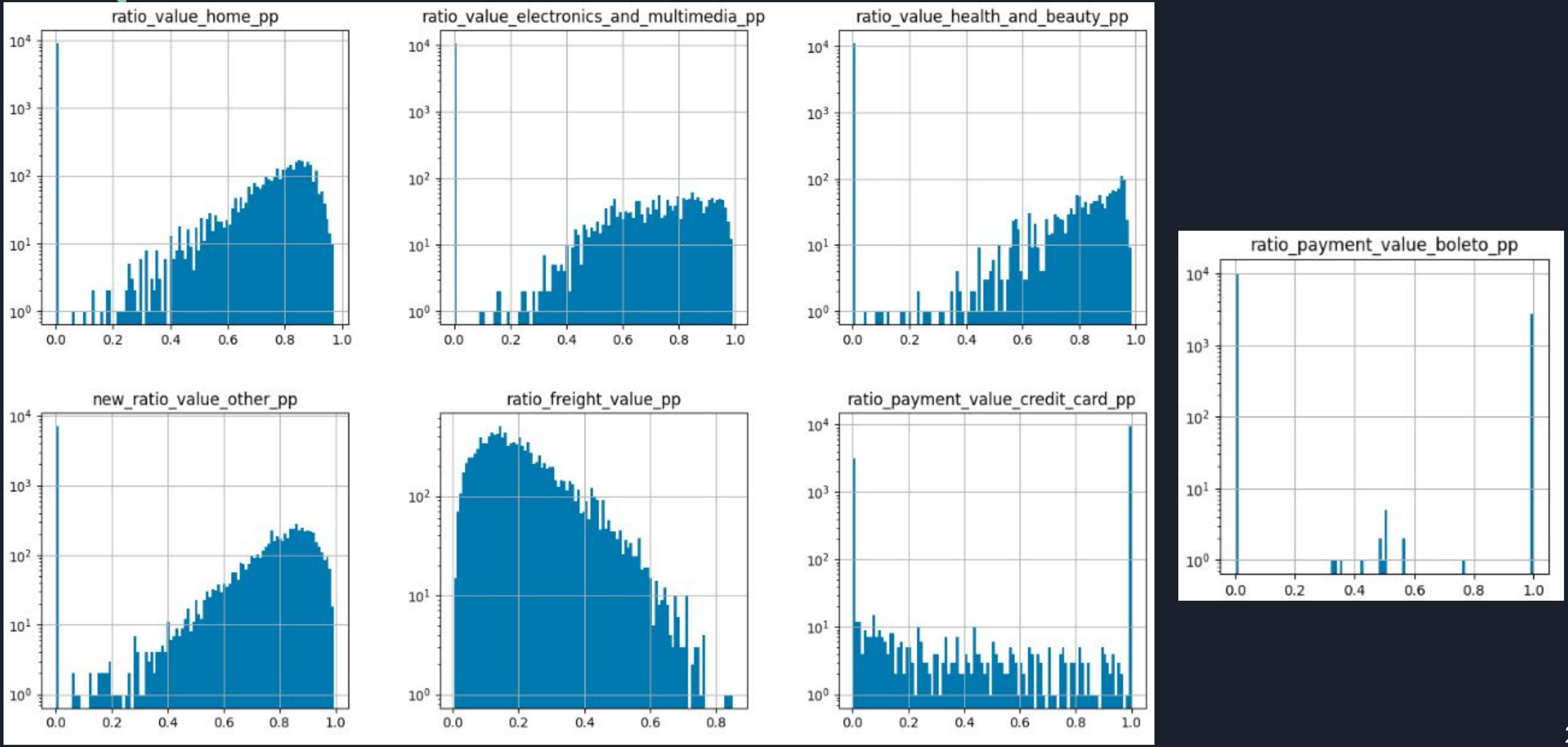
'Days since last purchase (recency)'
and
'Mean of the review scores'



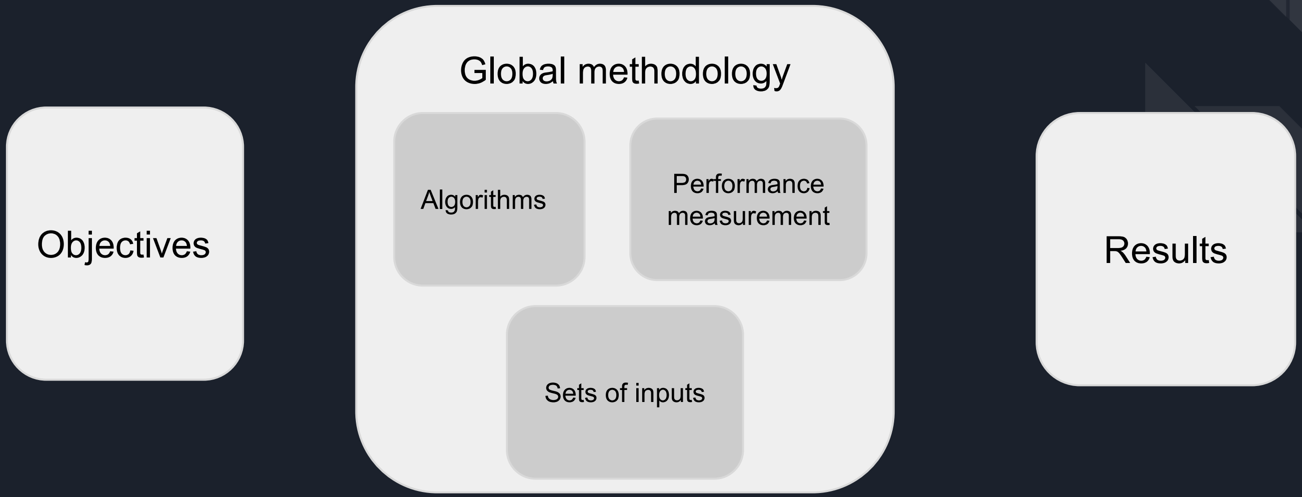
Standardization



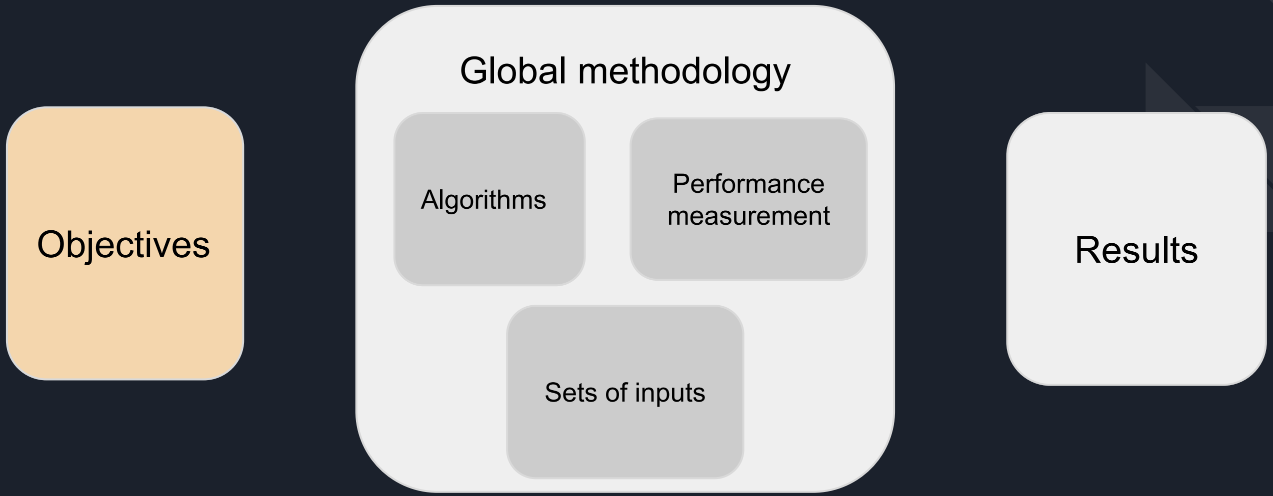
Others ratio values (no transformation)



3 - Methodology and results of the segmentation via unsupervised methods



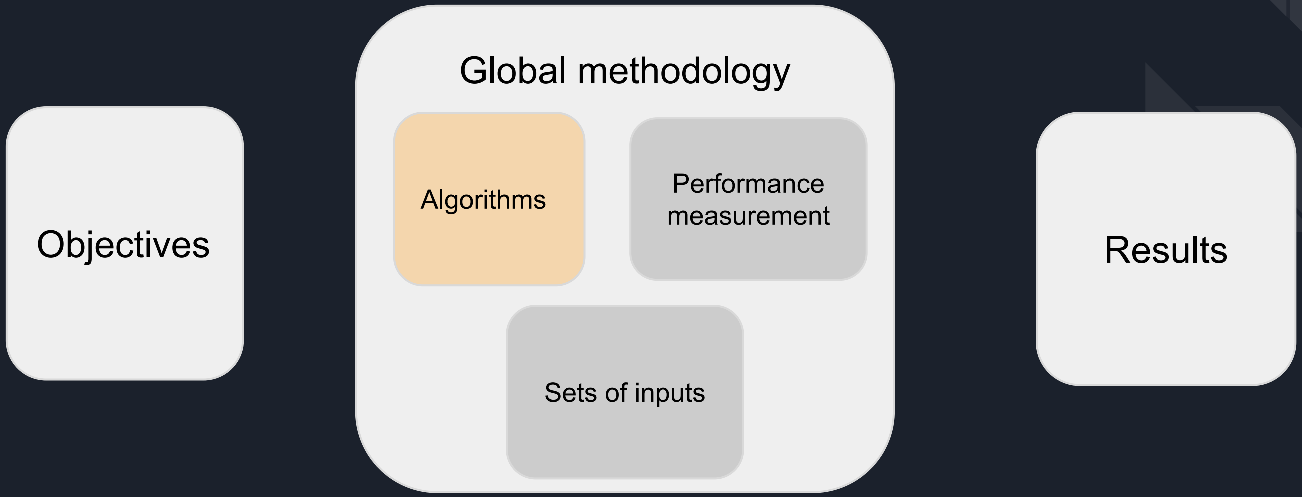
3 - Methodology and results of the segmentation via unsupervised methods



Requirements of the segmentation :

- actionable/meaningful to the marketing team.
- classify all the clients.
- distinguish small and important customers.
- distinguish satisfied and unsatisfied customers.

3 - Methodology and results of the segmentation via unsupervised methods



Algorithms

Name	Principle	Pros	Cons
KMeans	• Random initialization of centroids. • Iteratively attach points to a centroid minimizing the global inertia (within-cluster sum of squares) and recompute centroids.	• Fast • Do not need much memory • Centroids features can represent the cluster	• Need to pre-determine the number of clusters. • Random initialization can lead to different results.
Hierarchical clustering *(ward linkage criterion)*	• Initially consider each point as a cluster. • Iteratively group one point to its closer cluster (distance defined by the chosen linkage criterion.) • Stop when all clusters are grouped into one unique cluster.	• Dendogram (visualization of natural groups) • Can lead to different segmentations without recomputing • Deterministic results.	• Need much more memory and time to run thank KMeans.
DBSCAN	• Run through all points of the dataset and connect them together if they share spatial vicinity (defined per a radius epsilon and a minimum number of points to be in the sphere defined by epsilon.)	• Can detect noisy points • Can detect complex manifolds.	• 2 parameters to tune.
OPTICS		• Visualization of possible clusters thanks to the reachability plot.	

3 - Methodology and results of the segmentation via unsupervised methods

Objectives

Global methodology

- Algorithms
- Performance measurement
- Sets of inputs

Results

Mixing theory with the practical.

Theoretical approach

	Main idea	range	how to read it ?
Silhouette score	Compare mean distance of a point to : ● its cluster points. ● and the nearest cluster points.	[-1 ; 1]	Well-assigned when close to 1
Calinski-Harabasz index	between-clusters dispersion / within-cluster dispersion	positive	the greater, the better
Davies-Bouldin index	homogeneity / separability	positive	The closer to 0, the better.

Practical approach

- Cluster interpretability (boxplots per clusters per feature).
- Visualization in pca and t-SNE spaces.

3 - Methodology and results of the segmentation via unsupervised methods

Objectives

Global methodology

Algorithms

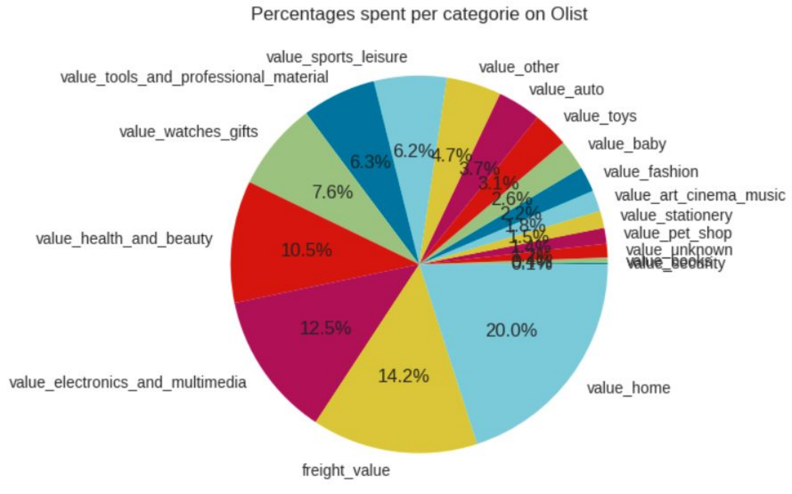
Performance measurement

Sets of inputs

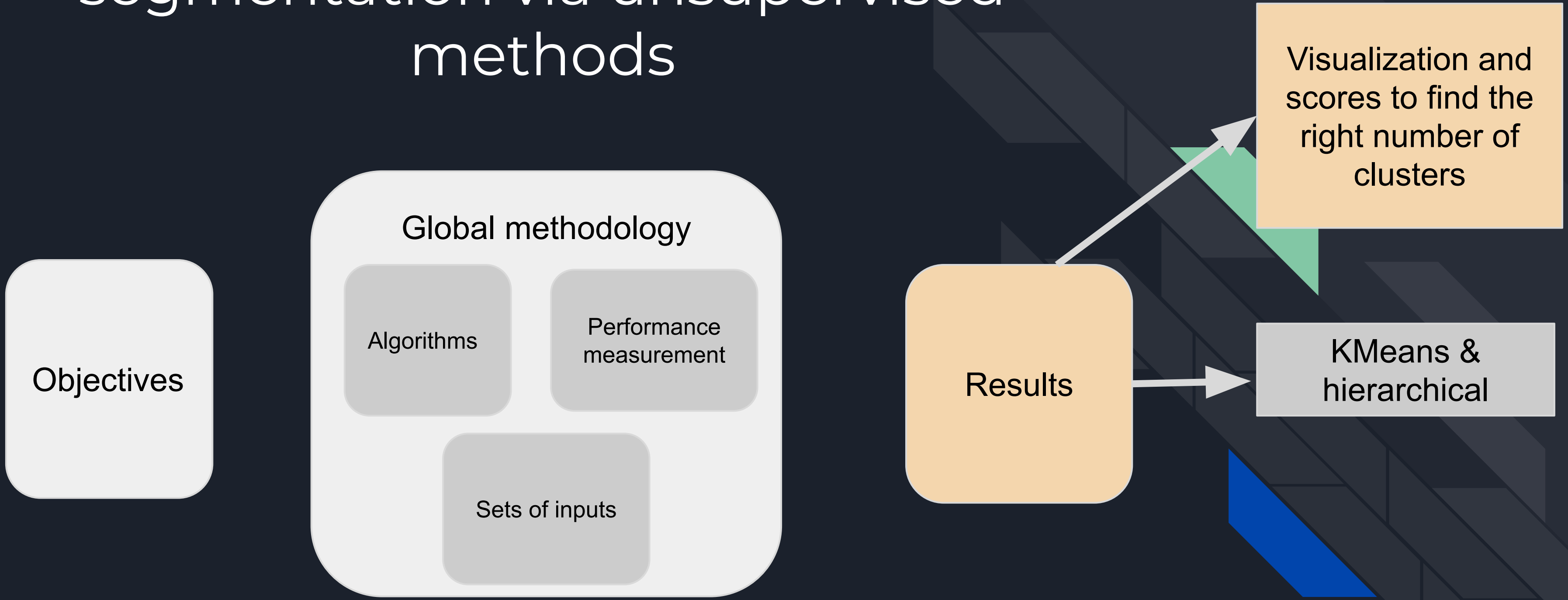
Results

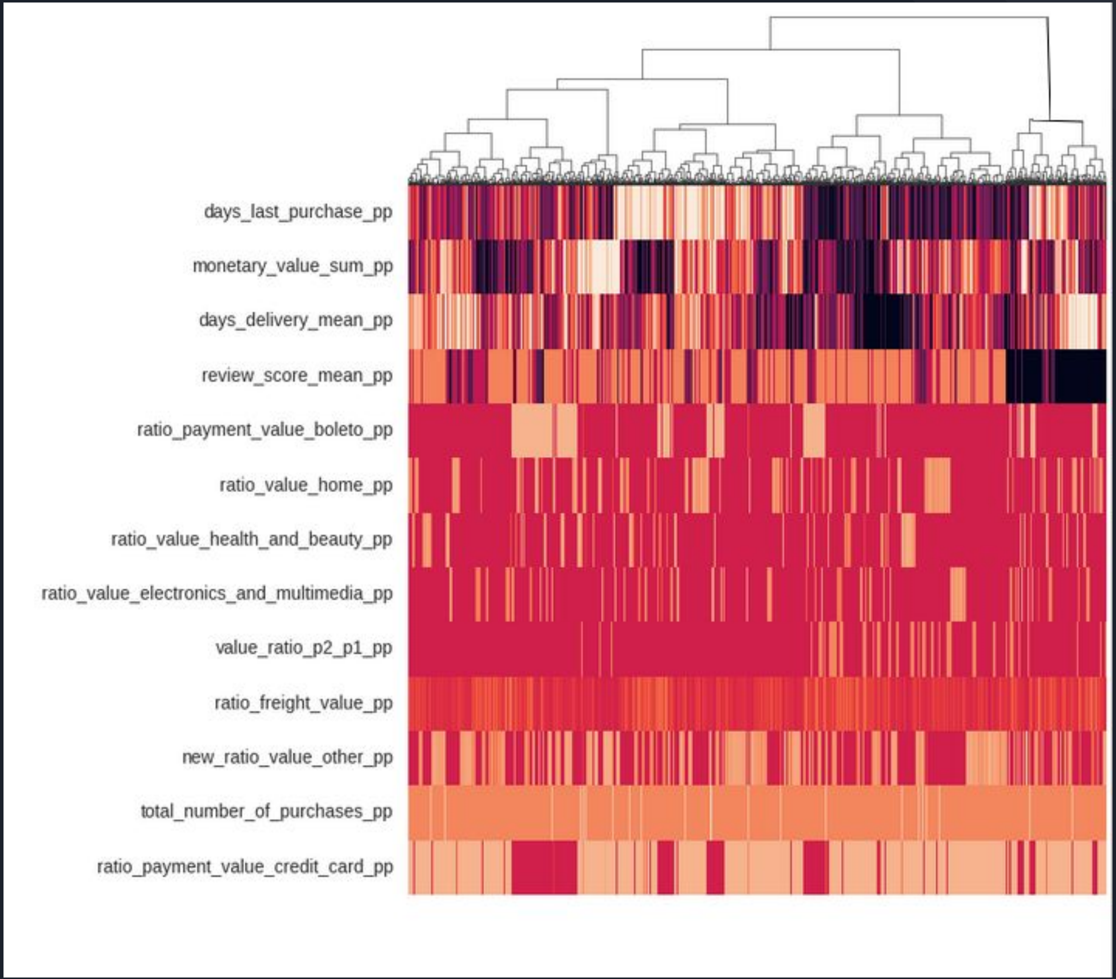
Sets of features used as inputs :

RFM set (3 features) ● Recency ● Frequency ● Monetary value	SET 2 (28 features) RFM set + mean review score + mean time delivery + dynamic ratio + mean number of items per order + All ratios values spent per categories of product) and per type of payment
SET 3 (13 features) Same as SET 2 but retrieved number of items and grouped all ratios from small categories into a new ratio.	SET 4 (6 features) SET 3 without ratios



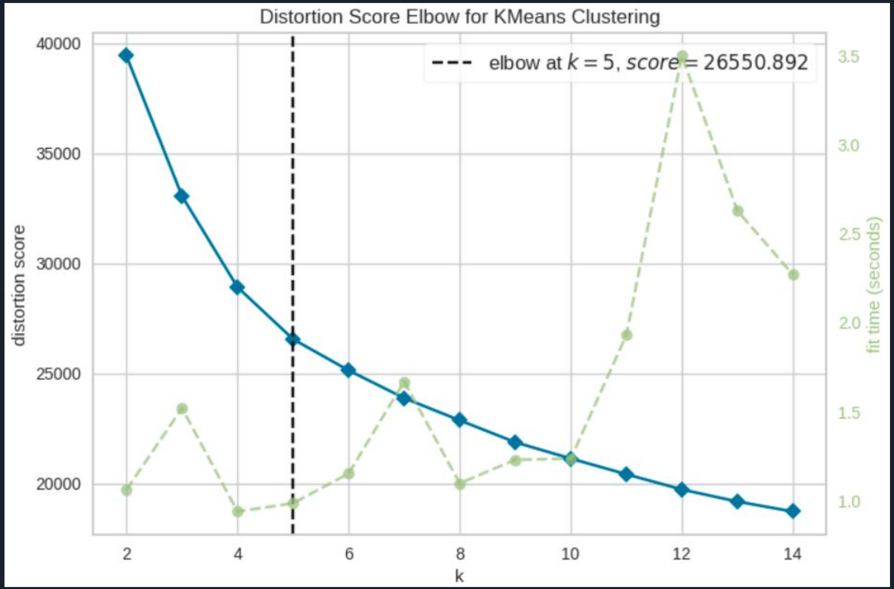
3 - Methodology and results of the segmentation via unsupervised methods

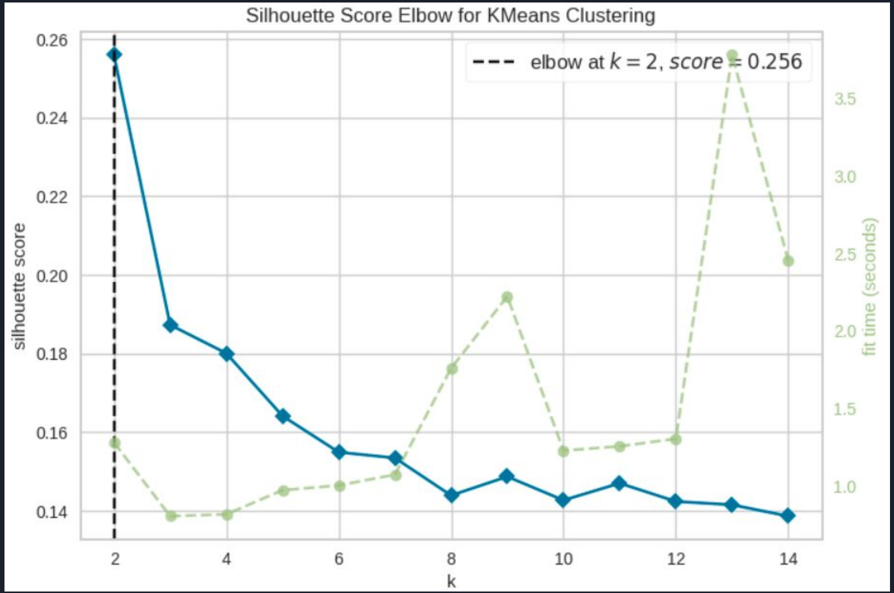




The dendogram suggests to use 4 or 5 clusters with the HAC (ward)

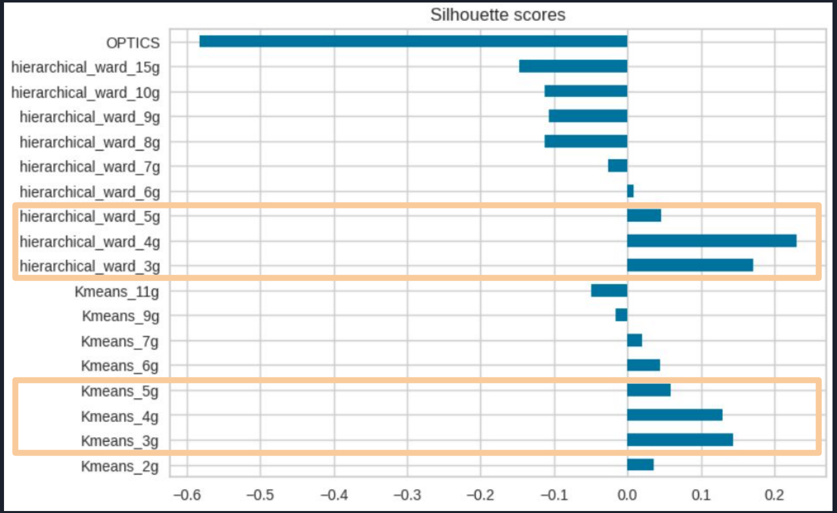
Choosing the number of clusters for KMeans using the elbow method and silhouette scores.

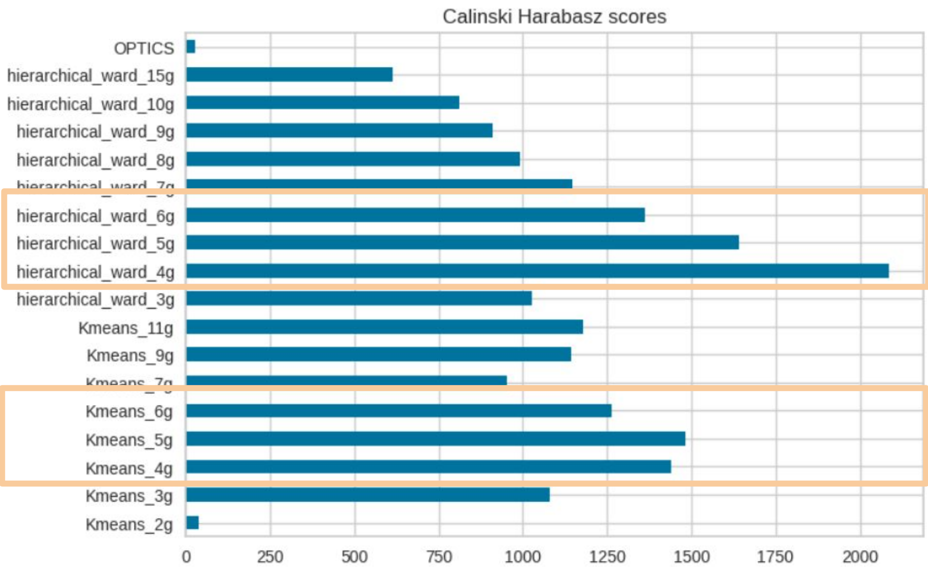


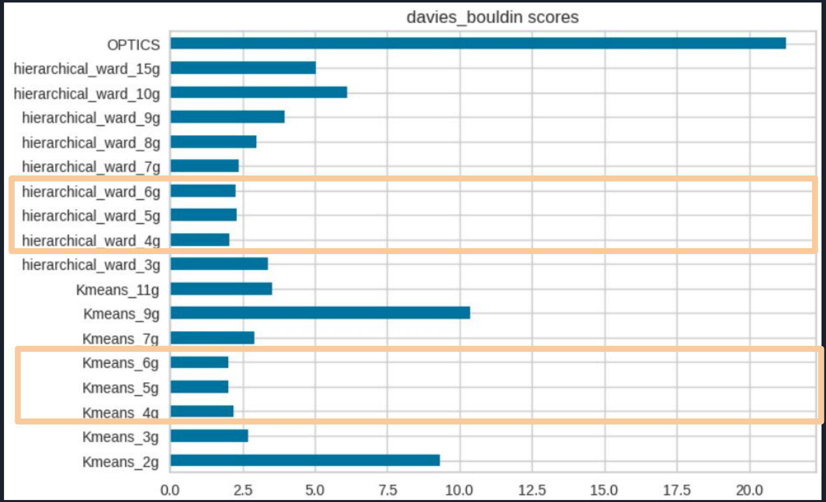


- Best silhouette score for 2 clusters, but it is not valuable for the marketing team.
- We want more clusters to distinguish several type of customers.
- There are no dramatic drop in silhouette scores → worth to explore for many k's around 5 (elbow method).

Scores with set 3 as input

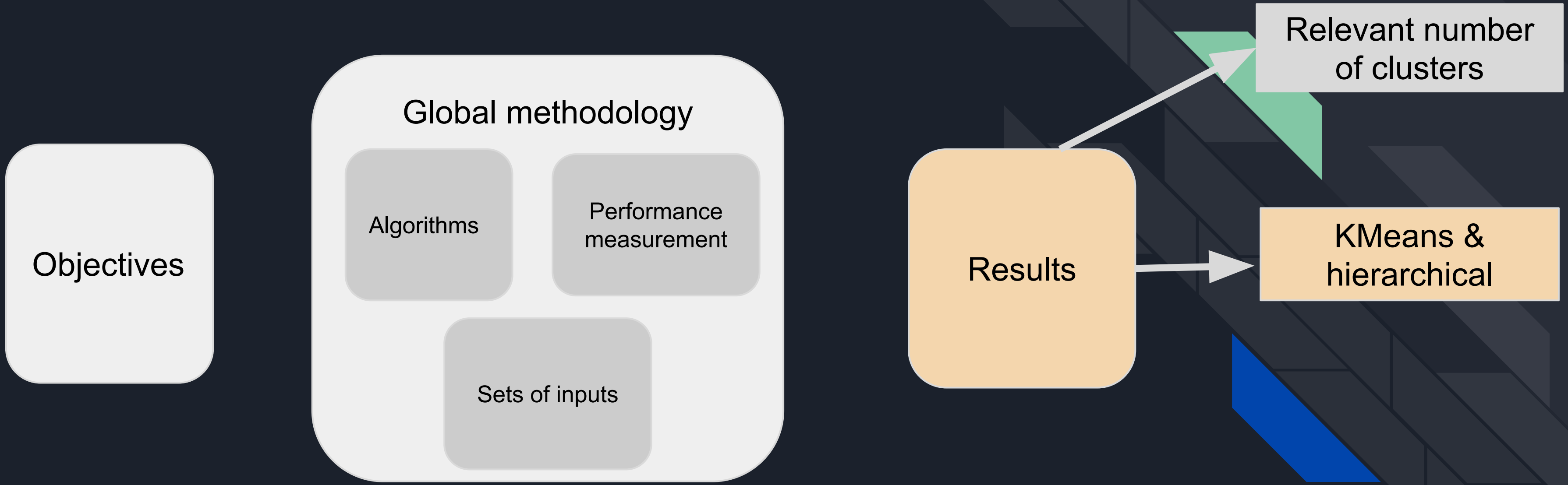






- OPTICS : always the worst. (possible to see why in appendices)
- Kmeans 4, 5, 6 groups : nice results.
- Same for Hierarchical 4, 5, 6 groups.

3 - Methodology and results of the segmentation via unsupervised methods

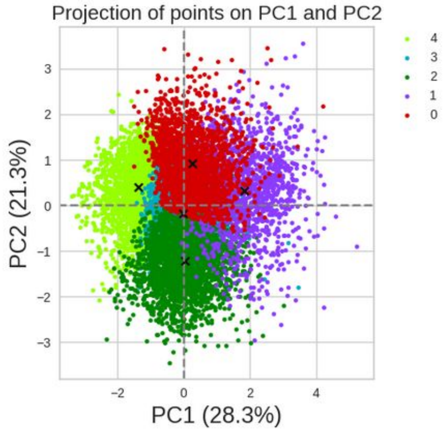


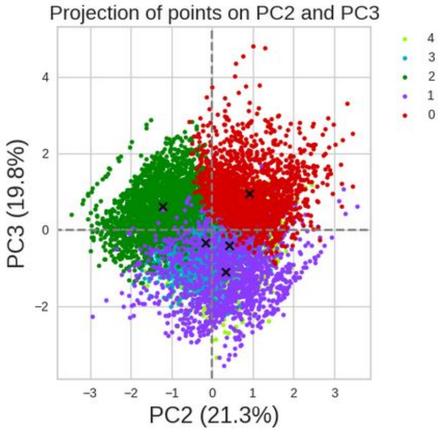
From a practical point of view :

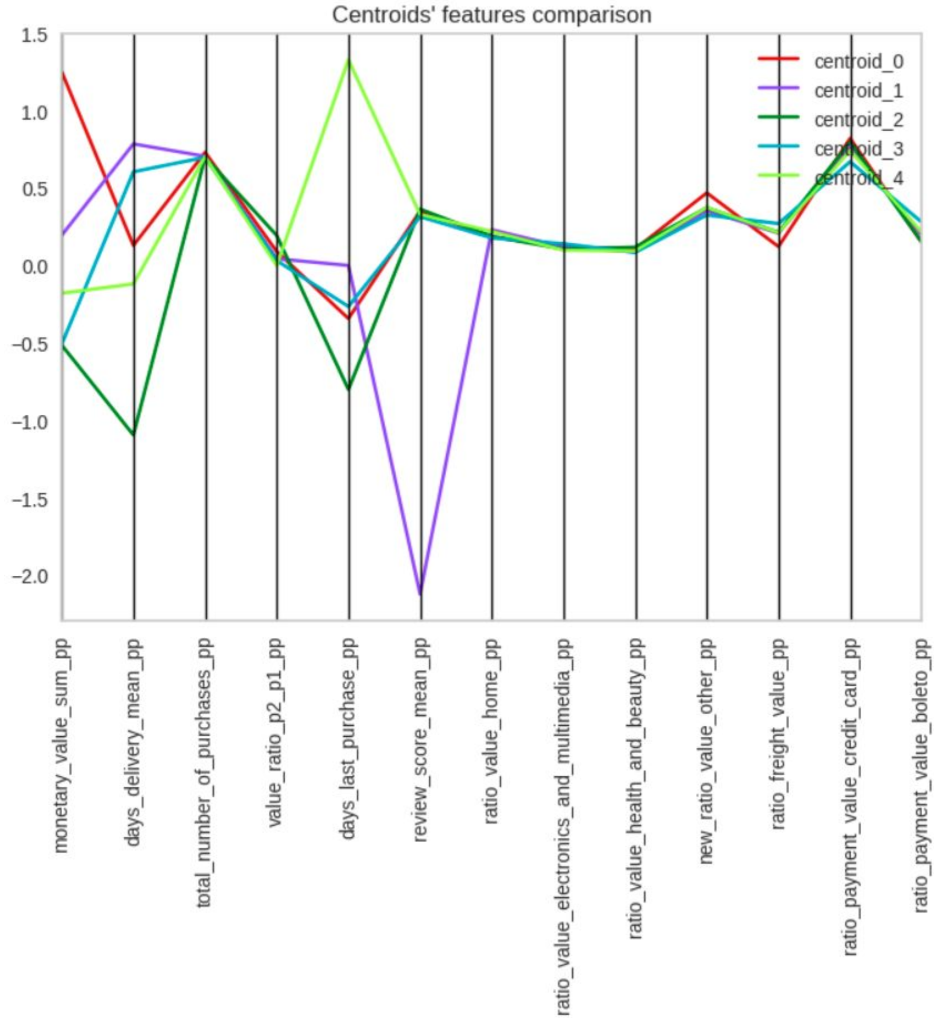
3 relevant segmentations detected :

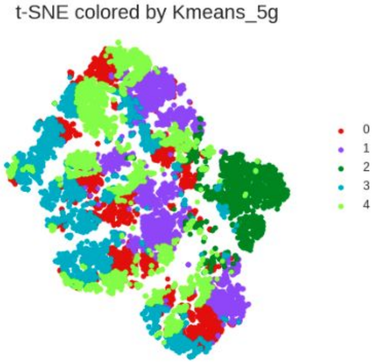
- Kmeans 5 groups
- Kmeans 7 groups
- Hierarchical clustering 4 groups

Kmeans - 5 groups

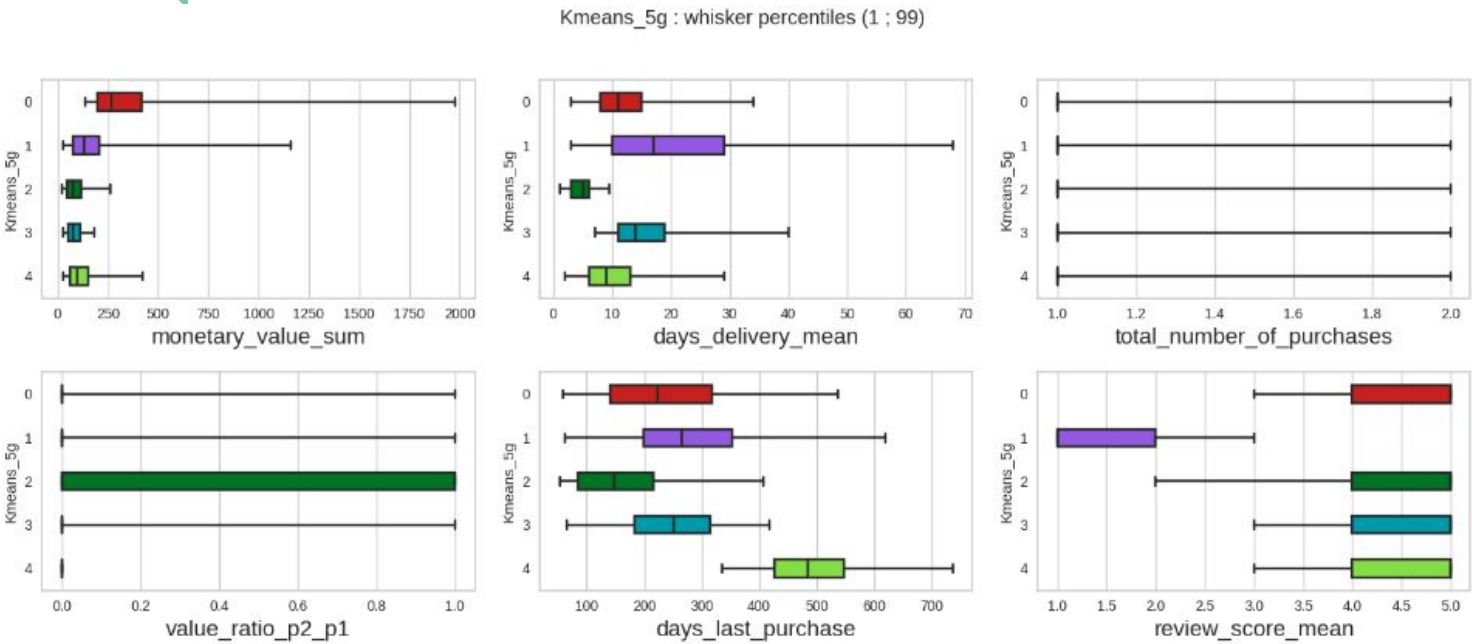








Kmeans - 5 groups



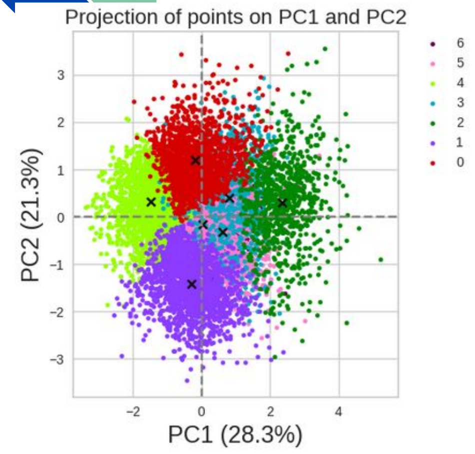
Clusters content :

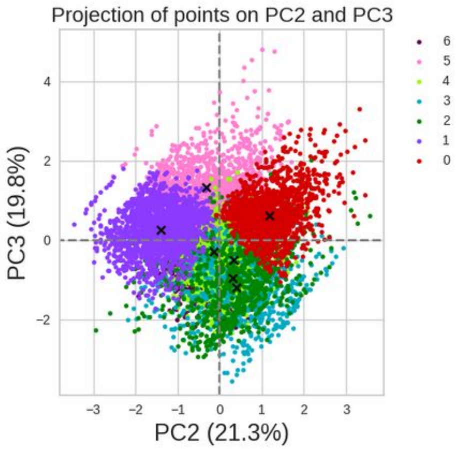
0 - Best customers (high value).
1 - Unsatisfied customers.
2 - Active customers (small values).
3 - On the verge to be less active (long delivery compared to 2).
4 - Small values and inactive customers.

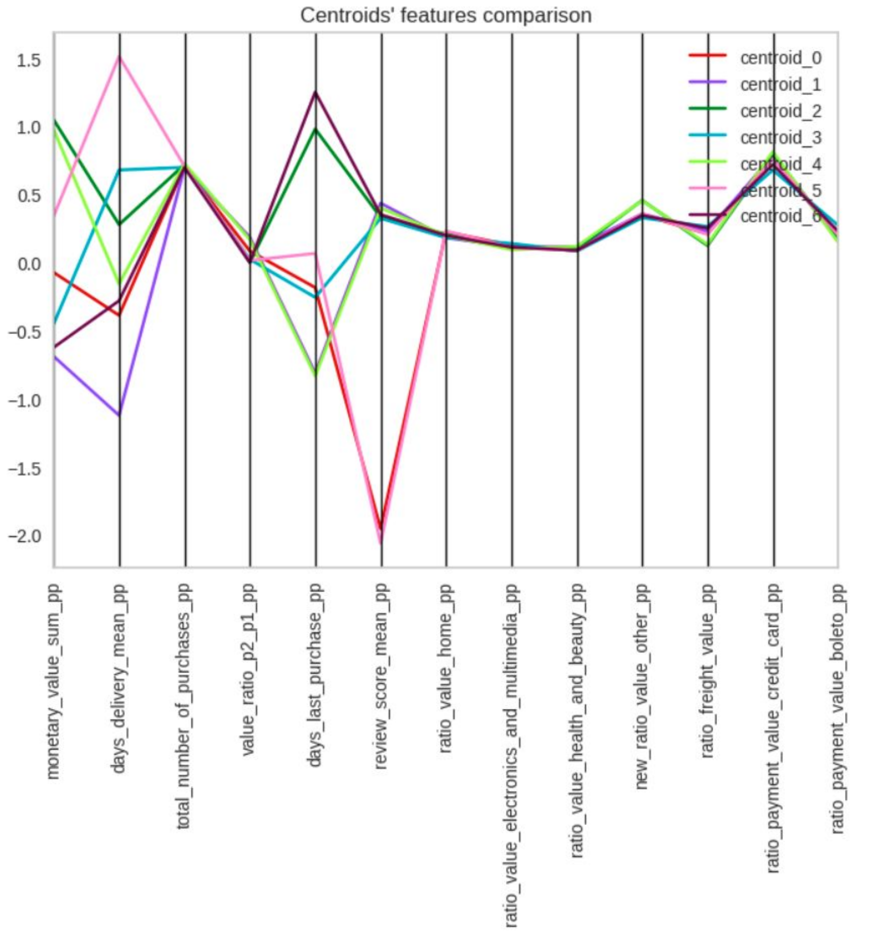
0, 1, 2 : credit card exclusivity for many customers.

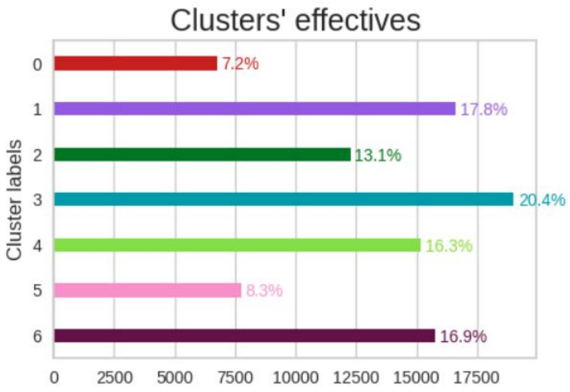
3, 4 : mix payment type

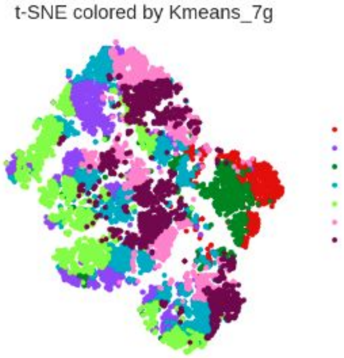
Kmeans - 7 groups



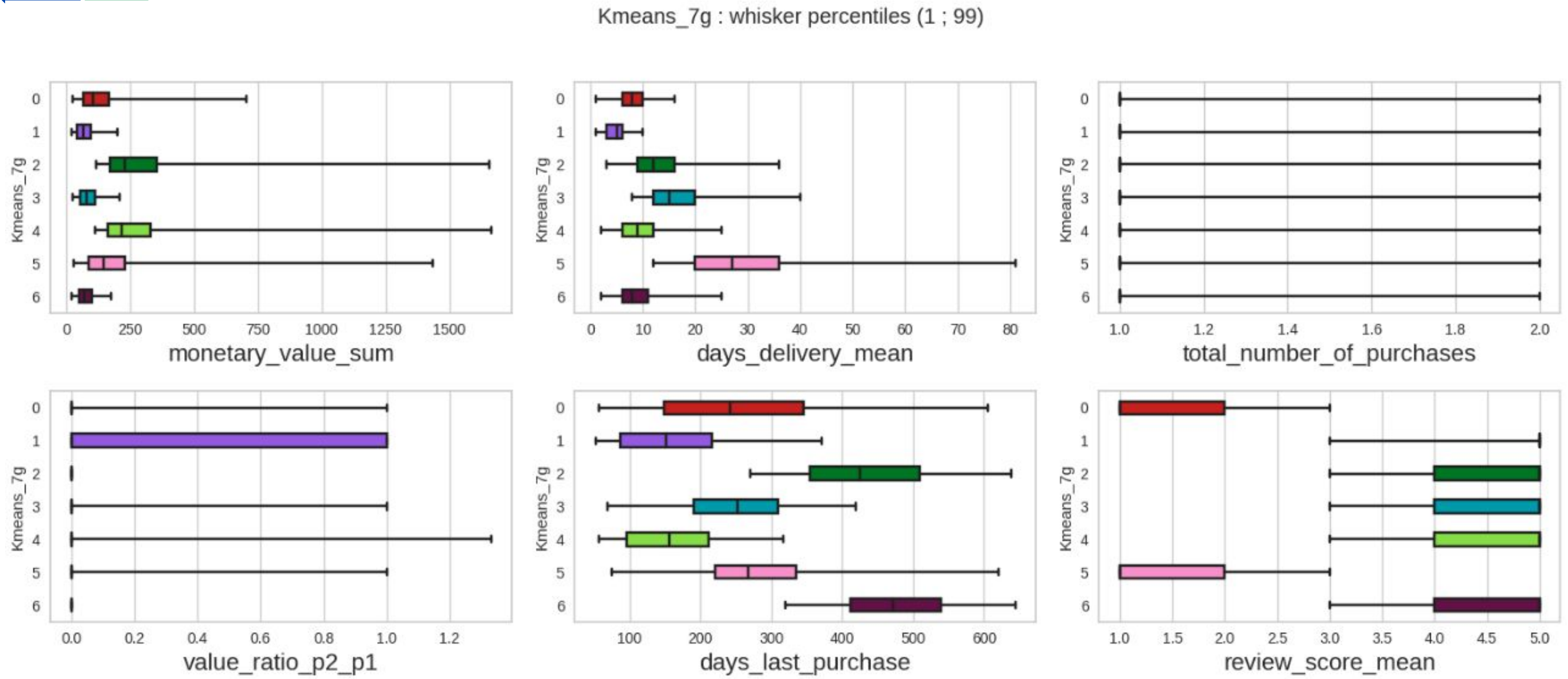






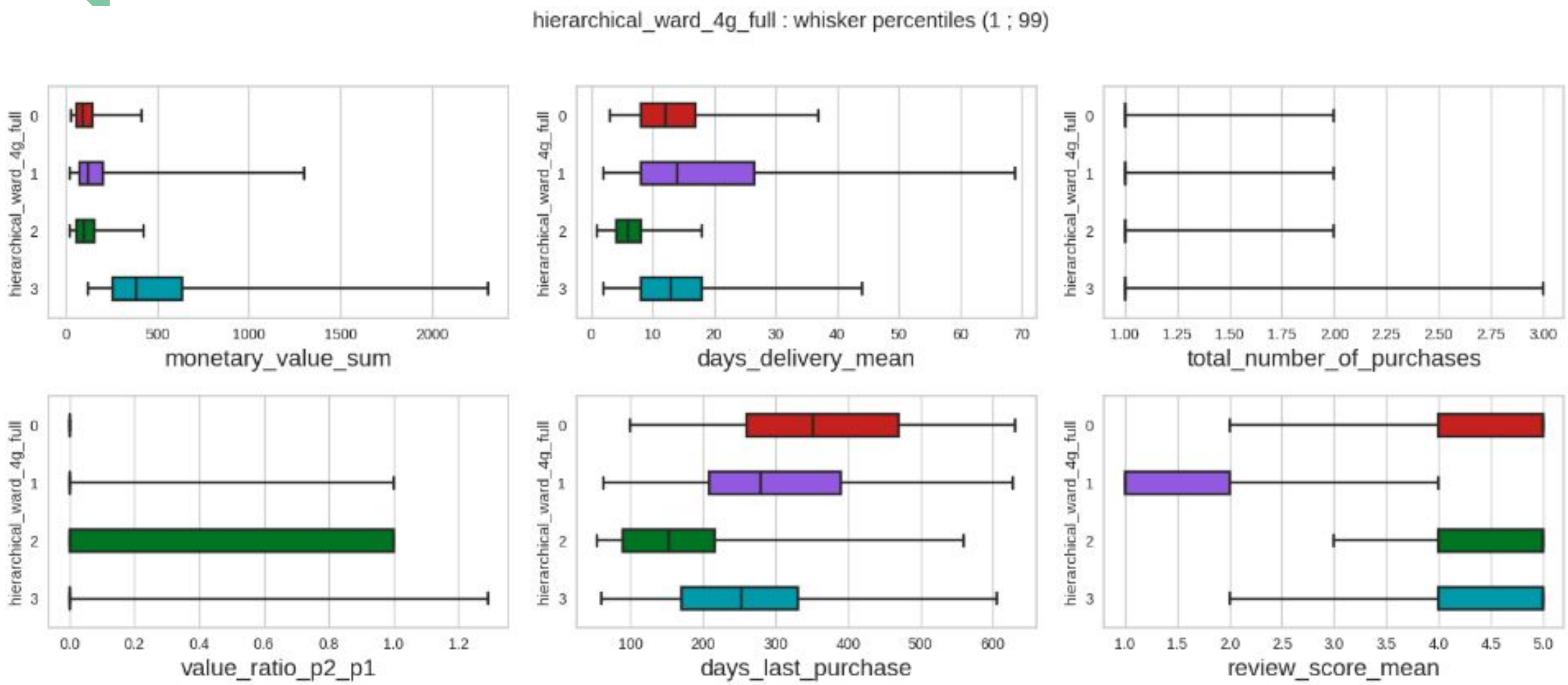


Kmeans - 7 groups



0 : Unsatisfied customers with short deliveries.
1 : Active customer, most satisfied, spend small values.
2 : Old inactive important customers.
3 : We are losing them. Spent small values, had a quite long delivery time.
4 : Recent important customers.
5 : Unsatisfied customers with long deliveries.
6 : Inactive customers who spent small values.

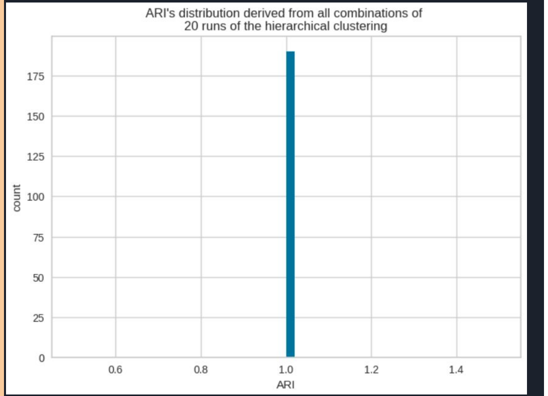
Hierarchical - 4 groups

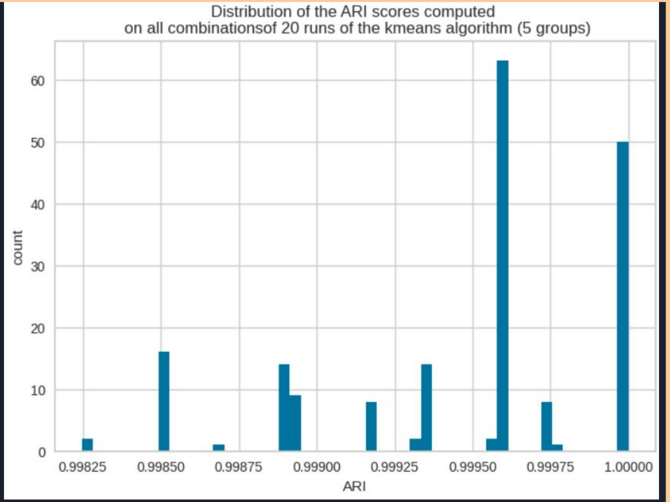


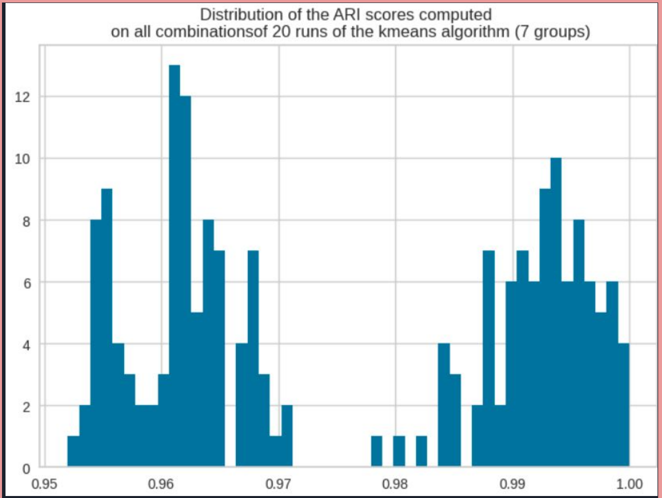
- Catches the minimal requirements.
- Increasing the group number in hierarchical clustering does not bring as much new useful information as for Kmeans.

Stability of those 3 segmentations on 10 000 customers.

- For each algorithm, run 20 times and store labels.
- Compute all combinations of Adjusted Rand Index between those labels and plot distributions to assess stability.







Very stable

Still a nice stability, but could start to introduce variability in cluster interpretability through runs ?

Segmentation conclusions

- OPTICS and DBSCAN are not adapted to the problem.

- Kmeans allow to segment clients on more relevant marketing criterions than the hierarchical clustering when we increase the number of clusters.

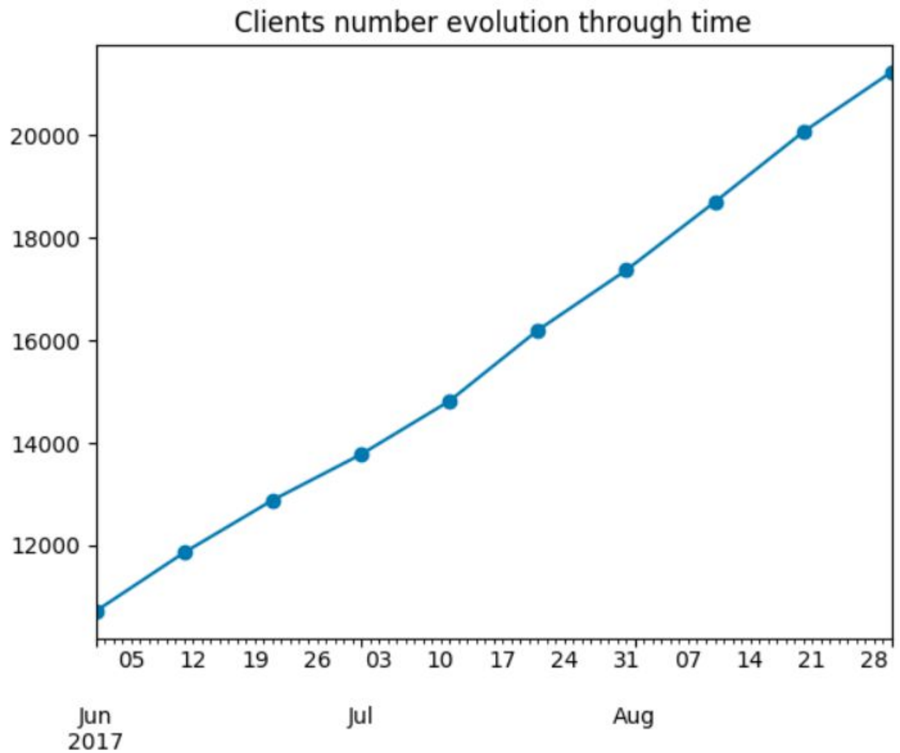
- The stability of the 3 winning algorithms is quite good but Kmeans 7 groups could start to introduce unstable customers compared to the other two. Beyond that, it remains the most interesting way to segment customers according to me. Thus, if the marketing team is willing to accept slightly less accuracy for more information, it is the segmentation to opt for.

- Kmeans 5 groups is safe and meet all criterions.

4 - Maintenance evaluation on Kmeans 5 groups.

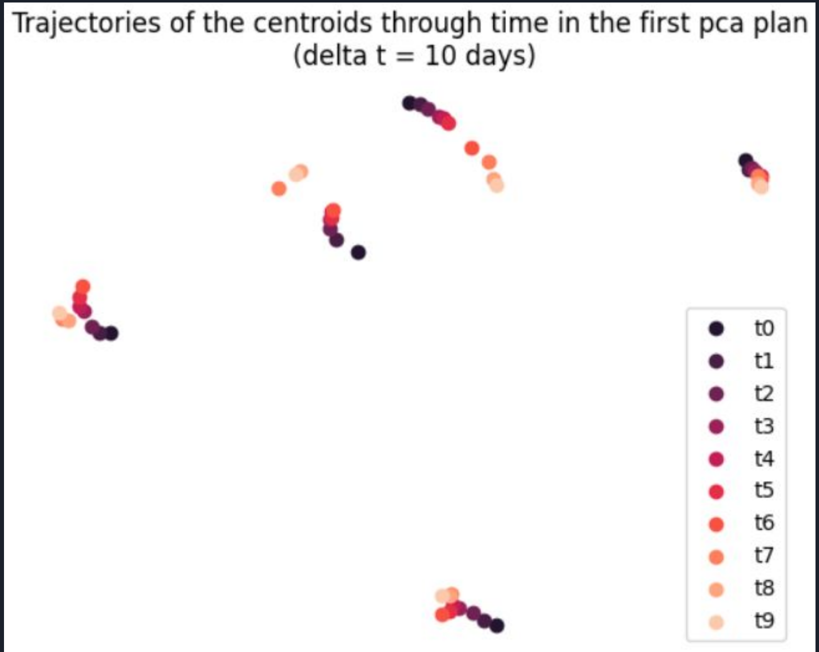
➔ Compute client summaries at 10 differents dates with a delta time of 10 days.

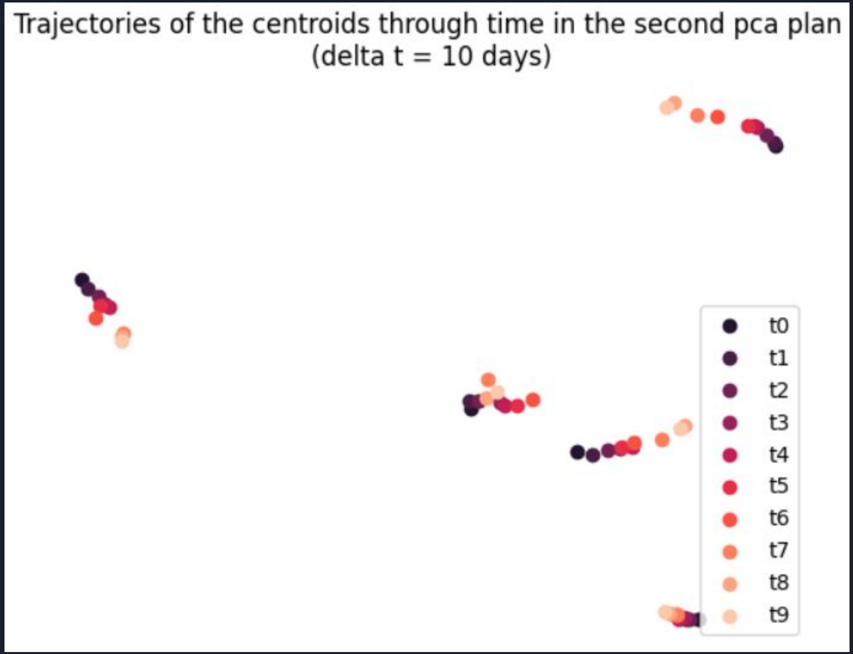
Client number evolution



- Mean number of new clients in 10 days in that period : 1 170.
- Mean number of new clients in 10 days in the last moment we possess data : 1812.

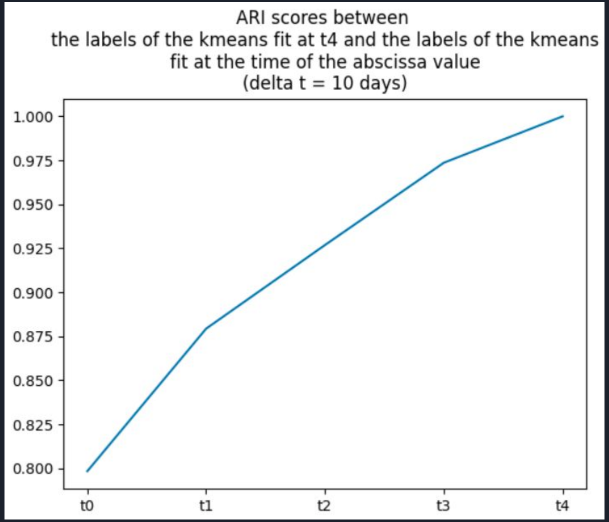
Centroids trajectories

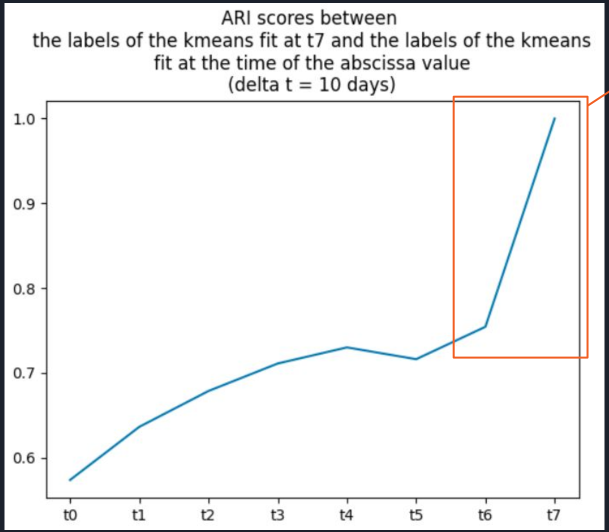


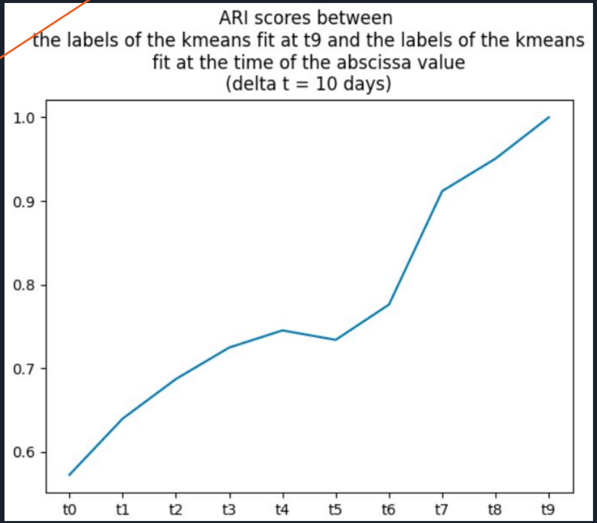


Centroids are quite stable, but they have a tendency to move and to never come back to previous position → Need of a maintenance.

ARI degradation through time when predicting with old models compared to the brand-new possible predictions.







If linear model , slope in the first graph : (0.2 / 4) = 0.05. In the last graph : (0.43/10) = 0.043.

0.05 can be considered as a mean loss of ARI per 10 days. Even though it can go much faster (middle graph).

Maintenance conclusions

- Each 10 days, around 1800 new clients will not be classified and will not be able to be targeted by the marketing team.
- The ARI score is also going down (0.05 each 10 days) leading to more and more misclassifications compared to an update.

→ Thresholding at an ARI of 0.85. The maintenance should be done each month. But 5400 new clients would be ignore during that period

Moreover, it sometimes decreases faster.

- I recommend an update each 10 days to be confident in the classification at every moment.
- It can remain quite relevant 1 month later though in a large number of situation.

Thanks for listening.

Appendices

RFM segmentation

recency_group	frequency_group	monetary_group	rfm_code
4	3	4	434
4	3	4	434
3	3	4	334
2	3	4	234
1	3	3	133
3	3	4	334
3	3	4	334
3	3	3	333
4	3	4	434

based on 3 features :

- Recency : days since last purchase
- Frequency : total number of purchase
- Monetary value : total value spent in all orders

PROS :

Easy to perform and intrinsically interpretable (actionable).

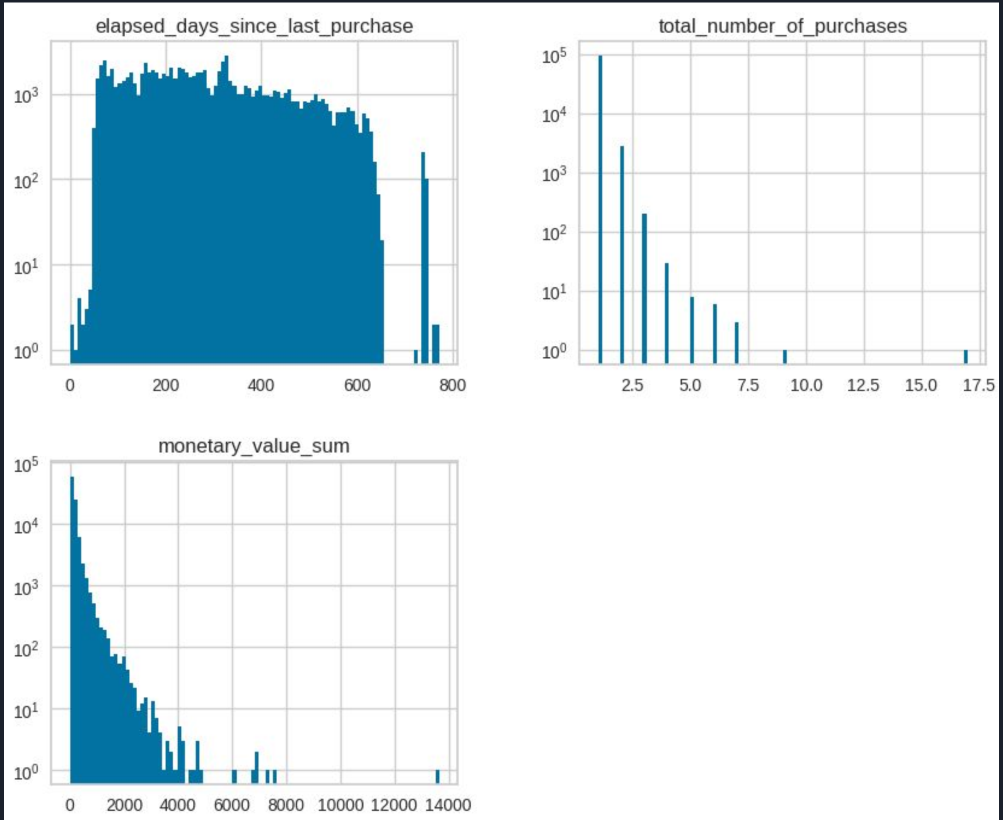
CONS :

- Does not take into account the dynamic.
- No information about client satisfaction (would introduce too many groups adding new features)

Idea :

→ assign a 3-digit-code classifying the customer (concatenate discretization step results).

My RFM Feature discretization



Recency :

- less than 90 days -> 1
- less than 180 days and not in previous -> 2
- less than 365 days and not in previous -> 3
- rest --> 4

Frequency :

- more than 5 -> 1
- 2,3 or 4 orders -> 2
- 1 order -> 3

Monetary:

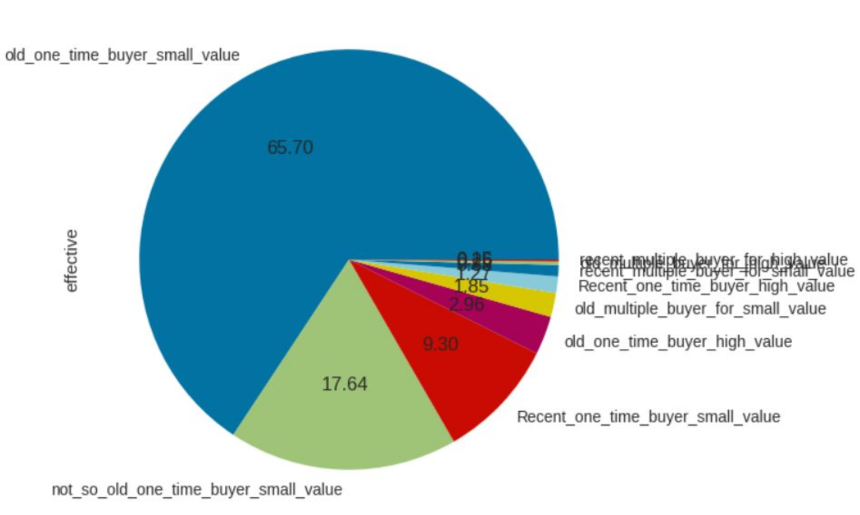
- more than ~500 dollars (2500 reales) -> 1
- between ~100 dollars and ~500 dollars (500 reales to 2499.99 reales) -> 2
- between 100 and 500 reales -> 3
- less than 100 reales -> 4

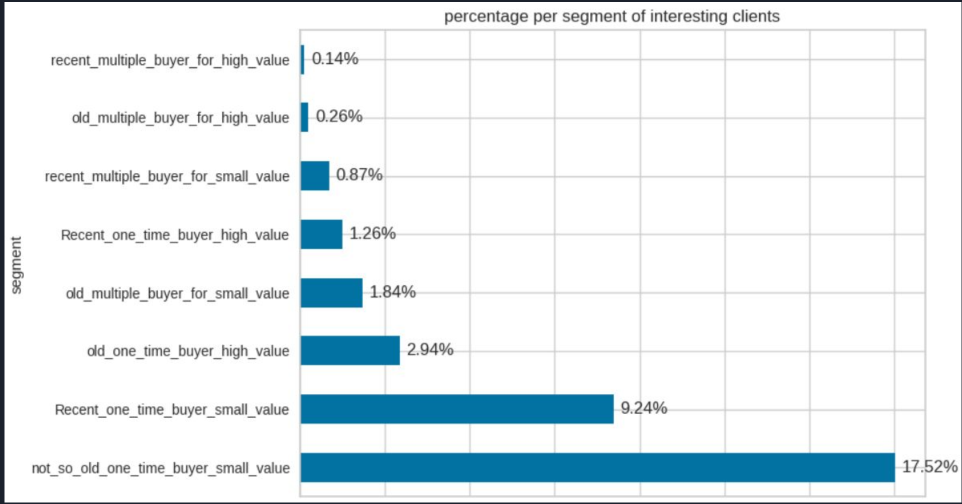
Thus, the higher the number for a feature, the worse it is for the platform.

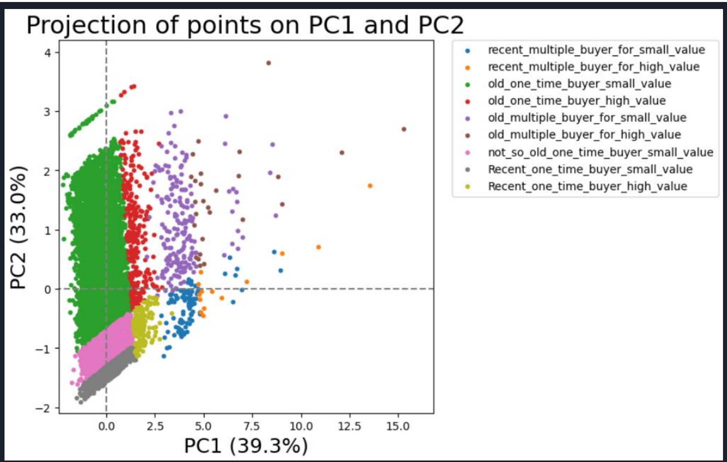
Mapping to understandable labels

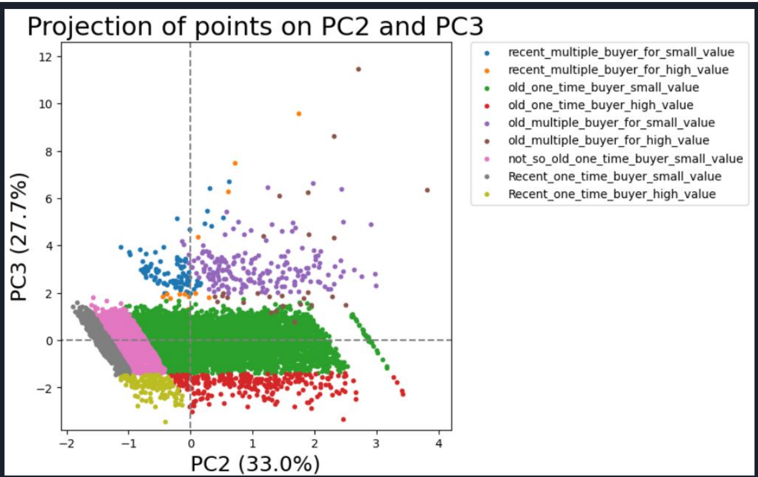
```
segment_map= {
    # One time buyers
    r'334|434|433|333': 'old_one_time_buyer_small_value',
    r'233|234': 'not_so_old_one_time_buyer_small_value',
    r'134|133': 'Recent_one_time_buyer_small_value',
    r'332|432|331|431': 'old_one_time_buyer_high_value',
    r'132|131|232|231': 'Recent_one_time_buyer_high_value',
    # With multiple orders
    r'[3-4][1-2][1-2]': 'old_multiple_buyer_for_high_value',
    r'[1-2][1-2][3-4]': 'recent_multiple_buyer_for_small_value',
    r'[1-2][1-2][1-2]': 'recent_multiple_buyer_for_high_value',
    r'[3-4][1-2][3-4]': 'old_multiple_buyer_for_small_value',
}
```

Results


old_one_time_buyer_small_value
65.70
effective
Recent_one_time_buyer_high_value
old_multiple_buyer_for_small_value
old_one_time_buyer_high_value
1.85
2.96
9.30
17.64
Recent_one_time_buyer_small_value
not_so_old_one_time_buyer_small_value


percentage per segment of interesting clients
segment
recent_multiple_buyer_for_high_value 0.14%
old_multiple_buyer_for_high_value 0.26%
recent_multiple_buyer_for_small_value 0.87%
Recent_one_time_buyer_high_value 1.26%
old_multiple_buyer_for_small_value 1.84%
old_one_time_buyer_high_value 2.94%
Recent_one_time_buyer_small_value 9.24%
not_so_old_one_time_buyer_small_value 17.52%


Projection of points on PC1 and PC2
PC1 (39.3%)
PC2 (33.0%)
recent_multiple_buyer_for_small_value
recent_multiple_buyer_for_high_value
old_one_time_buyer_small_value
old_one_time_buyer_high_value
old_multiple_buyer_for_small_value
old_multiple_buyer_for_high_value
not_so_old_one_time_buyer_small_value
Recent_one_time_buyer_small_value
Recent_one_time_buyer_high_value


Projection of points on PC2 and PC3
PC2 (33.0%)
PC3 (27.7%)
recent_multiple_buyer_for_small_value
recent_multiple_buyer_for_high_value
old_one_time_buyer_small_value
old_one_time_buyer_high_value
old_multiple_buyer_for_small_value
old_multiple_buyer_for_high_value
not_so_old_one_time_buyer_small_value
Recent_one_time_buyer_small_value
Recent_one_time_buyer_high_value

Addressing nulls in the merged dataset

Unknown because of the order status. If has not been shipped, no date... Let as such so each calculus based on this becomes a null too.

The order_status is 'canceled' or 'unavailable'. The information has consequently probably been deleted.
Let as such because not used in the features engineering.

One order. Corrected manually.

Each order_status leads to distinct review scores. I chose 4 when delivered and 2 when not.

replace by 'unknown'

customer_id	0
customer_unique_id	0
customer_zip_code_prefix	0
customer_city	0
customer_state	0
order_id	0
order_status	0
order_purchase_timestamp	0
order_approved_at	177
order_delivered_carrier_date	2086
order_delivered_customer_date	3421
order_estimated_delivery_date	0
order_item_id	833
product_id	833
seller_id	833
shipping_limit_date	833
price	833
freight_value	833
payment_sequential	3
payment_type	3
payment_installments	3
payment_value	3
total_order_cost	0
total_payment_value	0
cost_minus_payment	0
review_id	997
review_score	997
product_category_name	2542
product_category_name_english	2567
large_product_category	2567
binary_order_status	0

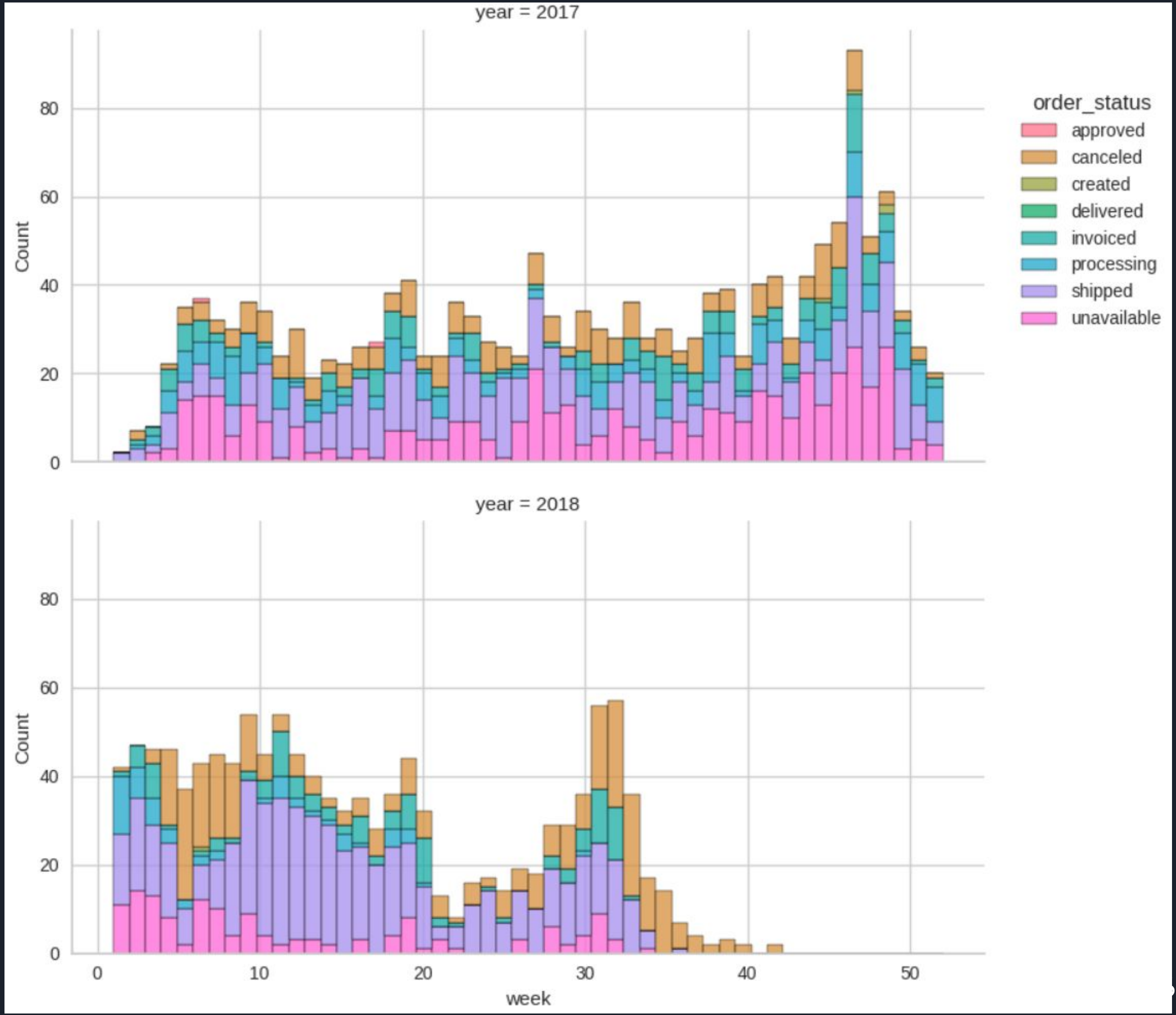
1st Oddity :

<u>My hypothesis :</u>

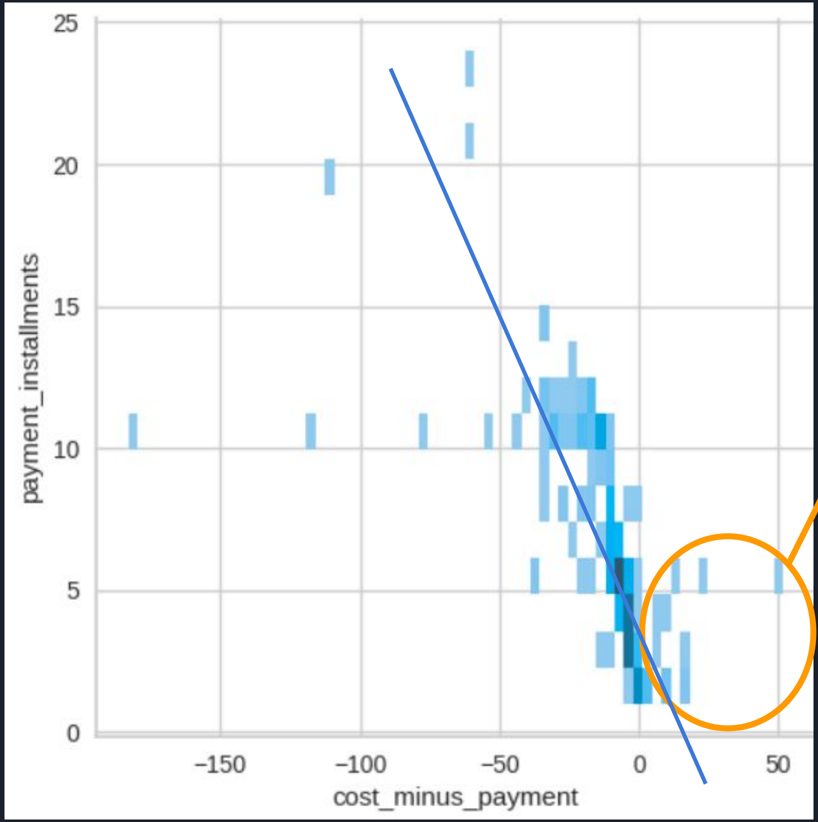
Order statuses such as 'approved', 'processing', or 'shipped' should only be encountered in the last week of the data set.

Else, they should have been changed to another status. 'canceled' or 'delivered'.

Status not updated ?



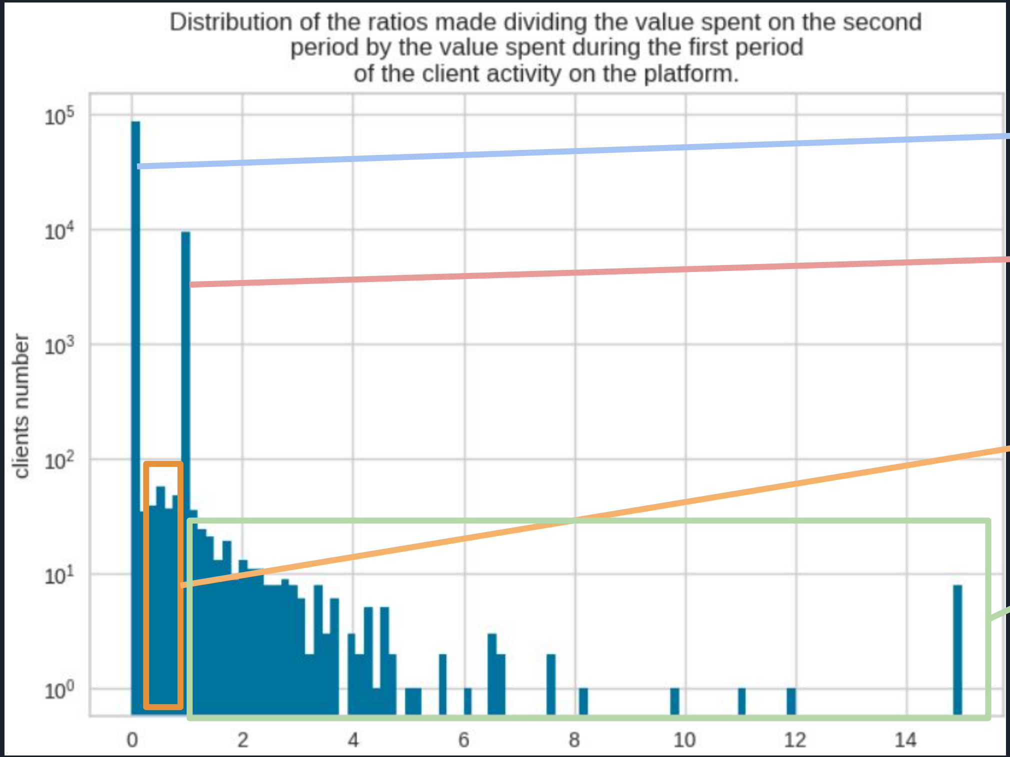
Second oddity among delivered orders



Differences between cost and payment

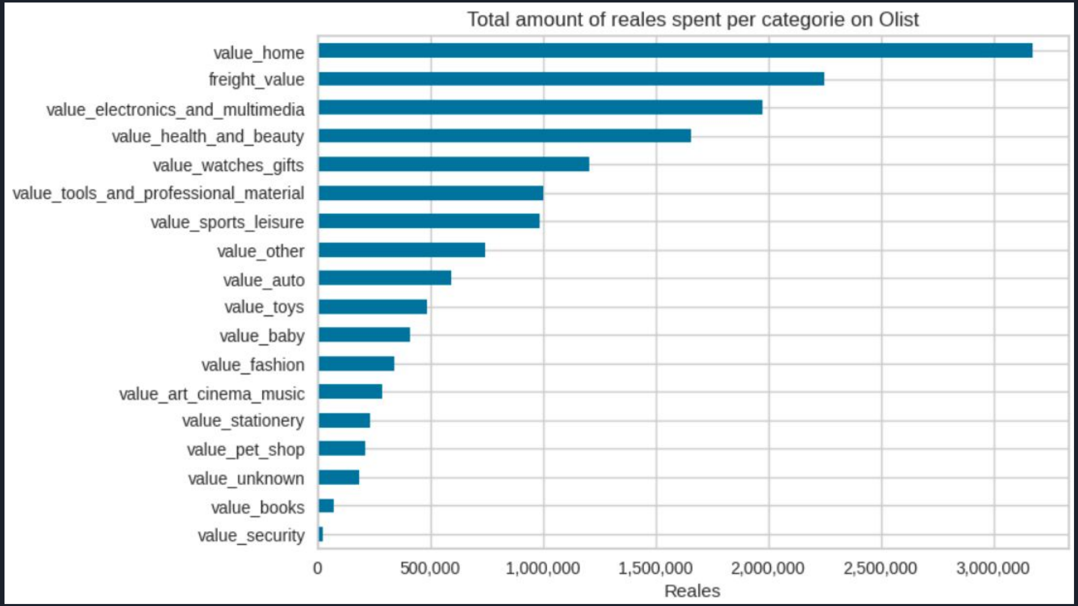
- We see that some clients paid less than due. (no explanation)
- Some paid more and we see a linear trend in function of the payment_installments. (additional fees not appearing in the data?)

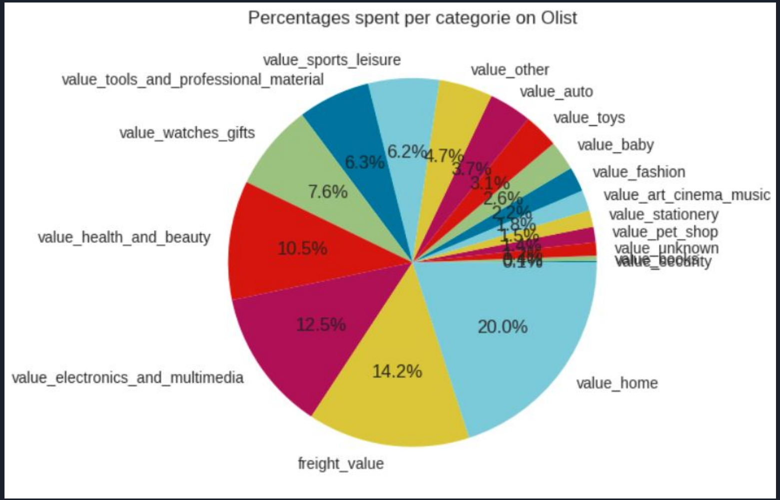
More on the dynamic ratio of the clients (y - log scale)

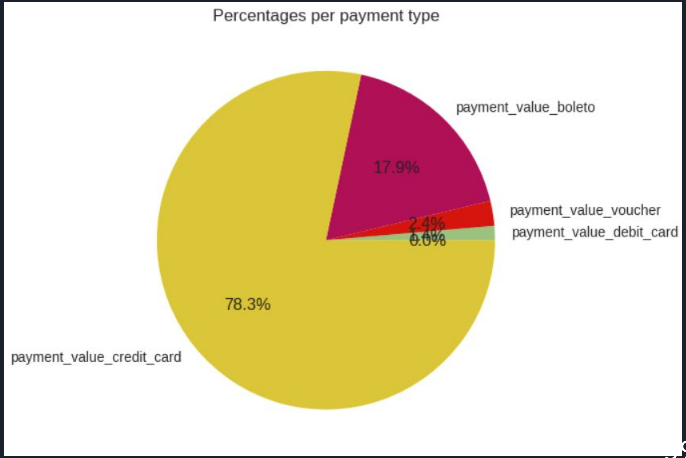


- Bought one time (more than 90 days ago).
- Bought one time (during the last 90 days) or spent exactly the same amount first and second half.
- Spent less in the second than in the first half of loyalty to the platform
- Spent more in the second than in the first half of loyalty to the platform

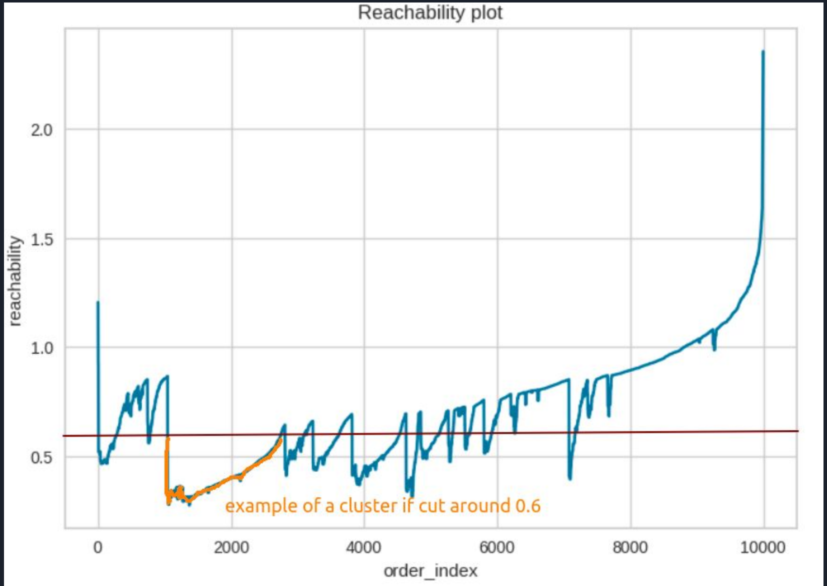
Incomes on the platform

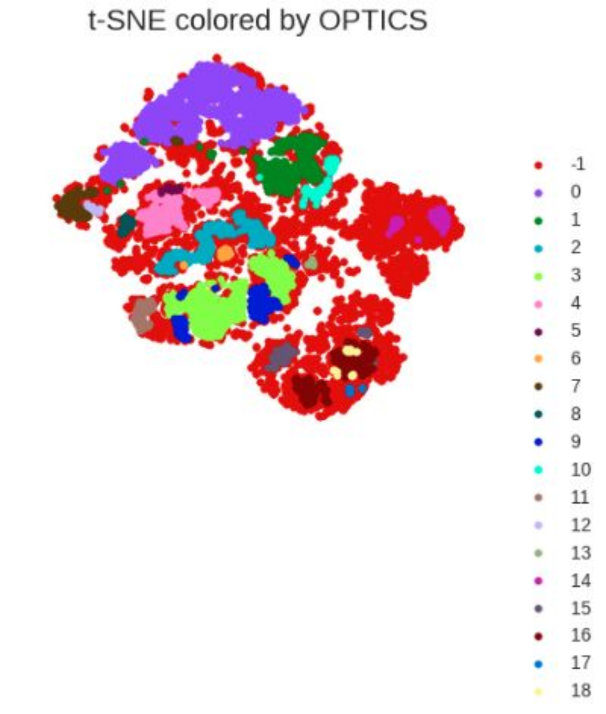






OPTICS and DBSCAN

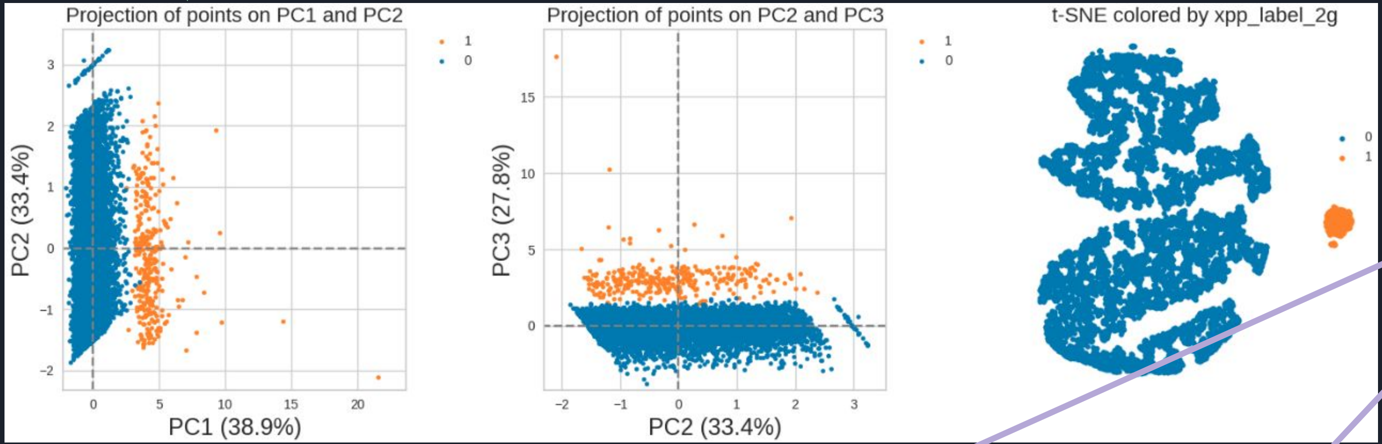




- Can not choose an epsilon on the y-axis to clearly separate points into a convenient number of clusters without considering a lot of points as noise (50% over the cut → red in the t-SNE)

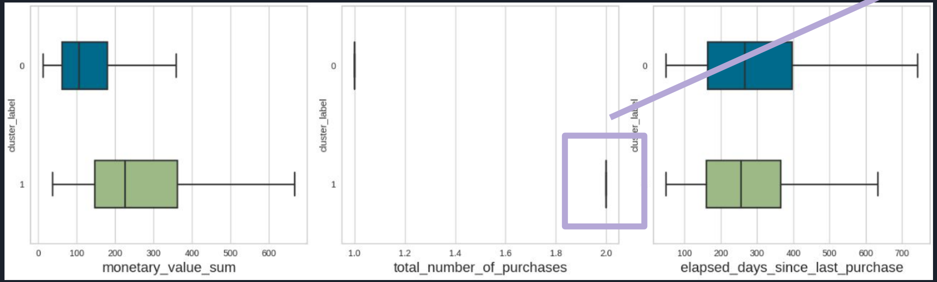
- NOT ADAPTED

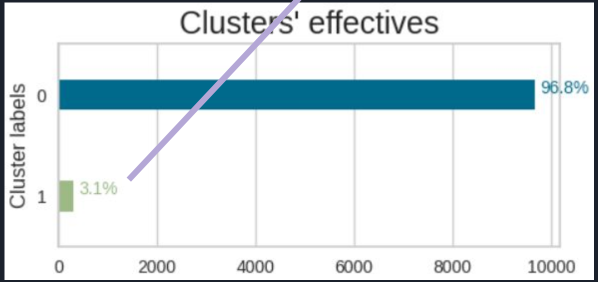
Results for k = 2 on the RFM set



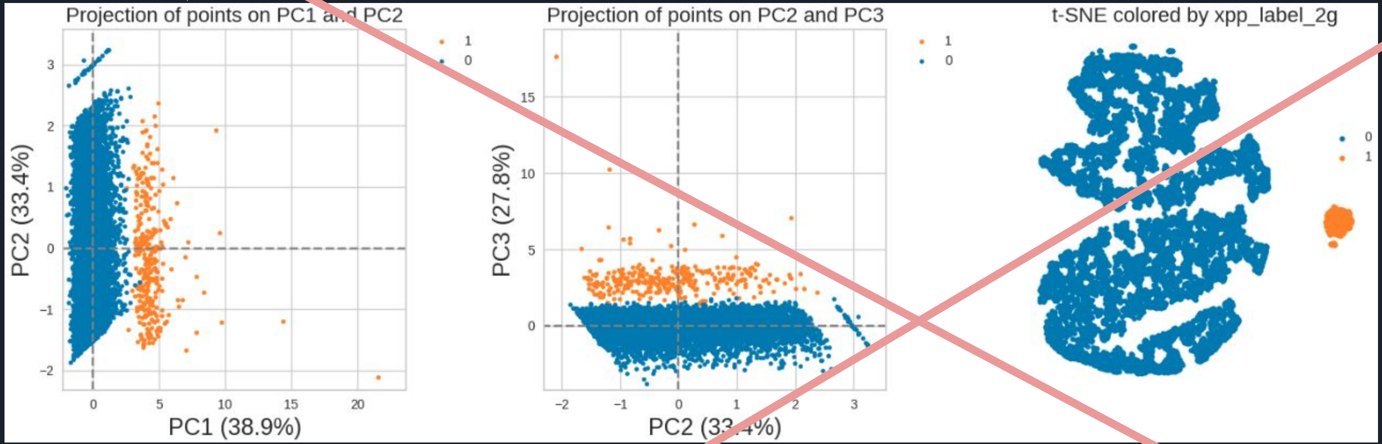
Help to label multiple-time buyers (3,1 %) .

Does not meet the requirements.

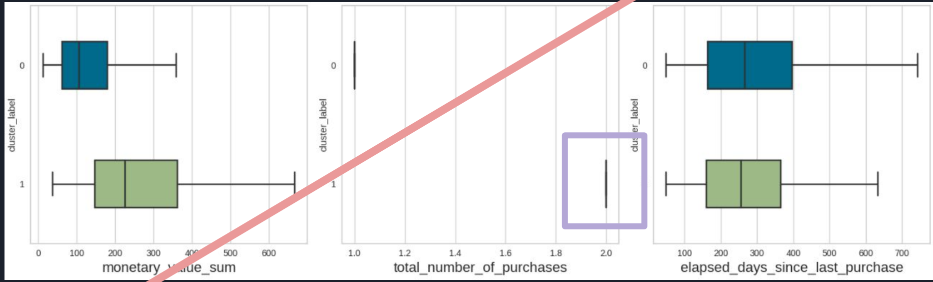




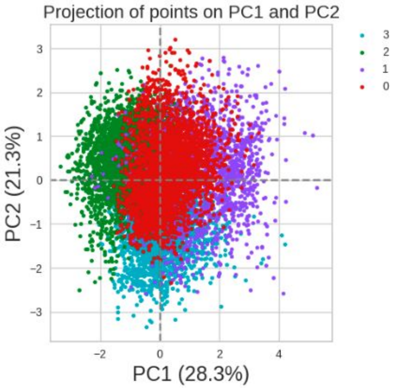
Results for k = 2 on the RFM set

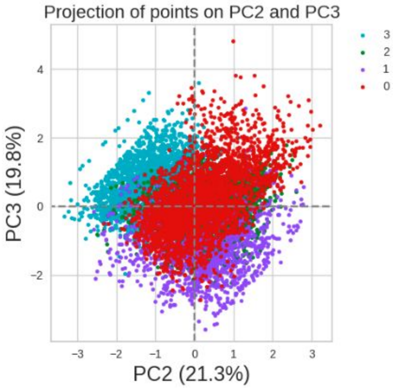


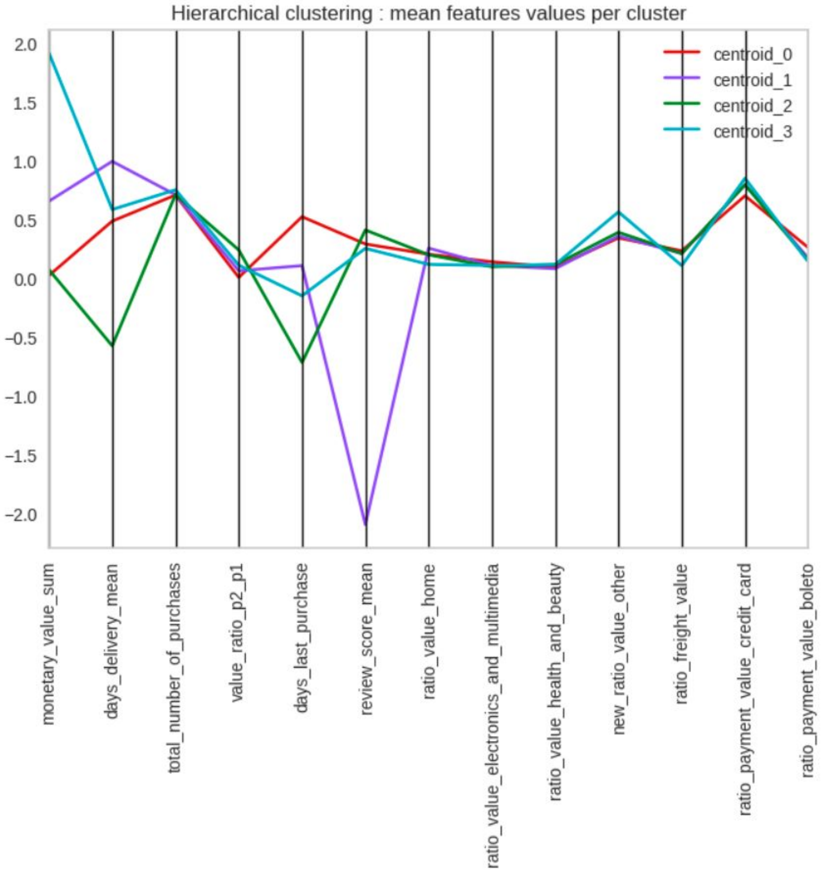
- Nothing about the satisfaction.
- Need more inputs.

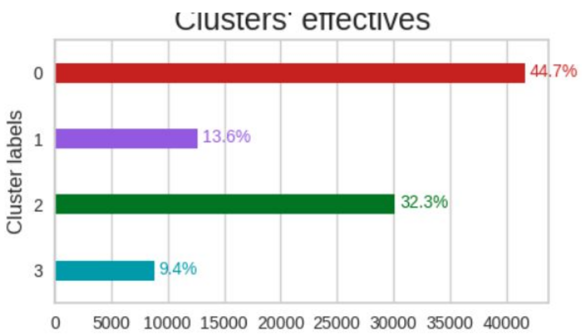


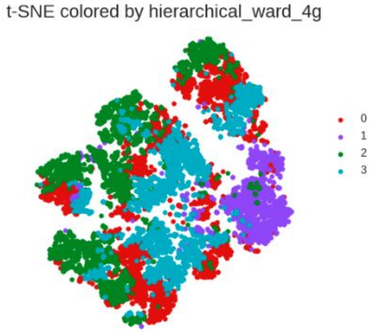
Hierarchical - 4 groups



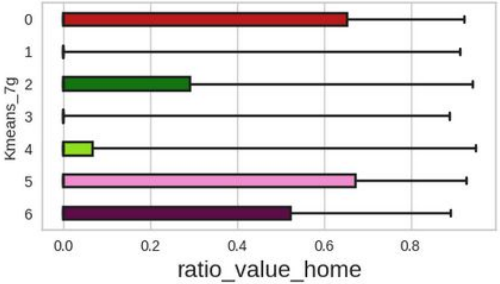


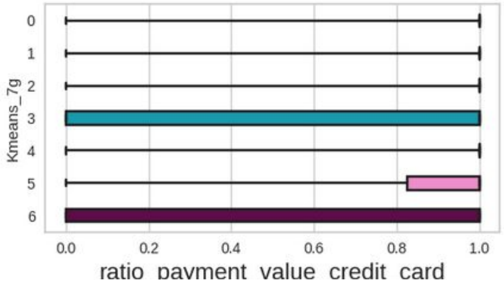


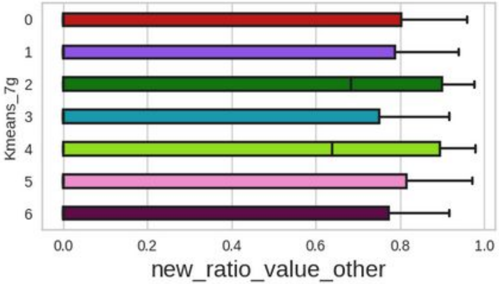


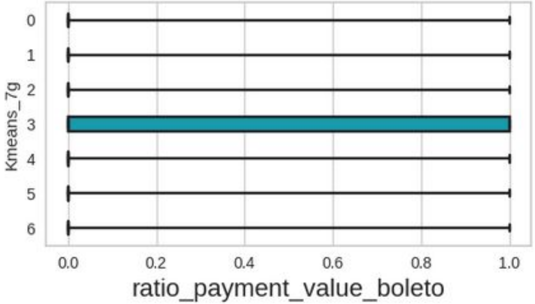


Kmeans - 7 groups (more boxplots and comment on groups)









0, 1, 2, 4 : credit card exclusivity (true for more than 75%)

3 : used a fair amount of banknotes.

5, 6 : used various payment type.

2, 4: lots of buyers in the resulting 'other category'.

0 : Unsatisfied customers with short deliveries.
1 : Active customer, most satisfied, spend small values.
2 : Old inactive important customers.
3 : We are losing them. Spent small values, had a quite long delivery time.

4 : Recent important customers.
5 : Unsatisfied customers with long deliveries.
6 : Inactive customers who spent small values.