PROJET N°5 :

SEGMENTEZ DES CLIENTS D'UN SITE E-COMMERCE

Soutenance du P5: le 12/06/2022

Version notebook : **6.3.0**
Version Python : **3.8.8**
Version Pandas : **1.2.4**
Version Seaborn : **0.11.1**
Version Matplotlib: **3.3.4**

Plan

- Contexte et présentation des Data-Set
- Traitement et nettoyage du Data-Set
- Analyse exploratoire
- Modélisation
- Modèles retenu et Stabilité
- Conclusion

Contexte et présentation des Data-Set

Contexte:

- Olist une entreprise brésilienne qui propose une solution de vente sur les marketplaces, nous fournit une base de données anonymisée comportant des informations sur l'historique de commandes, les produits achetés, les commentaires de satisfaction, et la localisation des clients depuis janvier 2017.
- L'objectif est de comprendre les différents types d'utilisateurs grâce à leur comportement et à leurs données personnelles anonymisées via des méthode des classifications non supervisées.

Mission:

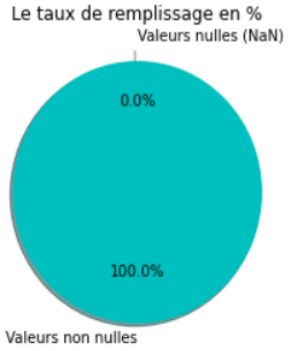
- Fournir aux équipes d'e-commerce Olist une description et une segmentation des clients actionnable qu'elles pourront utiliser au quotidien pour leurs campagnes de communication ainsi que une exploitation optimale.
- Proposition de contrat de maintenance basée sur une analyse de la stabilité des segments au cours du temps

Contexte et présentation des Data-Set

Présentation des tables de données:

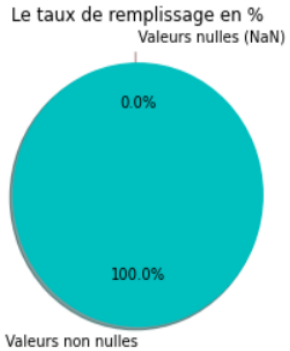
Customer

```
* Nombre de colonnes sans NaN -------: 5
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 0
* Nombre de lignes ------------------------: 99441
* Nombre de colonnes ----------------------: 5
* Nombre de cases -------------------------: 497205
* Nombre de valeurs nulles ----------------: 0
* Nombre de valeurs non nulles ------------: 497205
* le pourcentage des valeurs nulles ------: 0.0 %
* le pourcentage des valeurs non nulles --: 100.0 %
```



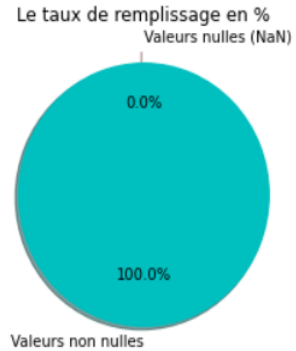
Geolocation

```
* Nombre de colonnes sans NaN -------: 5
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 0
* Nombre de lignes ------------------------: 1000163
* Nombre de colonnes ----------------------: 5
* Nombre de cases -------------------------: 5000815
* Nombre de valeurs nulles ----------------: 0
* Nombre de valeurs non nulles ------------: 5000815
* le pourcentage des valeurs nulles ------: 0.0 %
* le pourcentage des valeurs non nulles --: 100.0 %
```



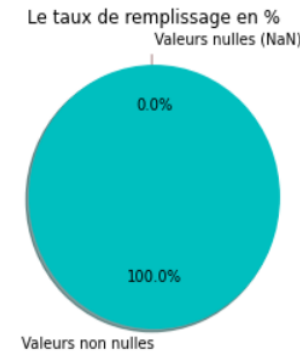
Order_items

```
* Nombre de colonnes sans NaN -------: 7
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 0
* Nombre de lignes ------------------------: 112650
* Nombre de colonnes ----------------------: 7
* Nombre de cases -------------------------: 788550
* Nombre de valeurs nulles ----------------: 0
* Nombre de valeurs non nulles ------------: 788550
* le pourcentage des valeurs nulles ------: 0.0 %
* le pourcentage des valeurs non nulles --: 100.0 %
```



Order_payments

```
* Nombre de colonnes sans NaN -------: 5
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 0
* Nombre de lignes ------------------------: 103886
* Nombre de colonnes ----------------------: 5
* Nombre de cases -------------------------: 519430
* Nombre de valeurs nulles ----------------: 0
* Nombre de valeurs non nulles ------------: 519430
* le pourcentage des valeurs nulles ------: 0.0 %
* le pourcentage des valeurs non nulles --: 100.0 %
```

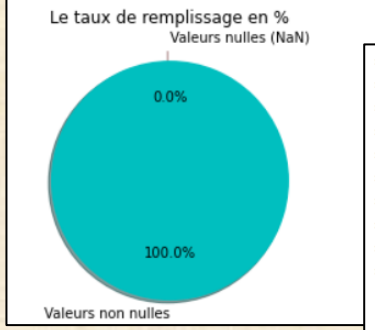


Contexte et présentation des Data-Set

Présentation des tables de données:

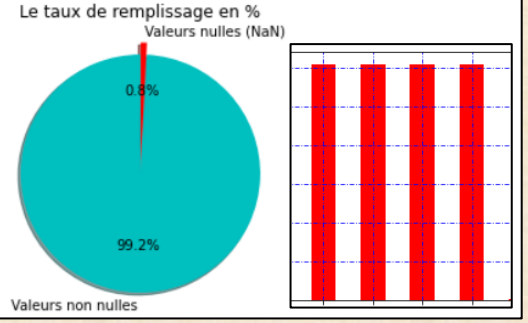
Sellers

```
* Nombre de colonnes sans NaN -------: 4
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 0
* Nombre de lignes ------------------------: 3095
* Nombre de colonnes ----------------------: 4
* Nombre de cases -------------------------: 12380
* Nombre de valeurs nulles ----------------: 0
* Nombre de valeurs non nulles ------------: 12380
* le pourcentage des valeurs nulles ------: 0.0 %
* le pourcentage des valeurs non nulles --: 100.0 %
```



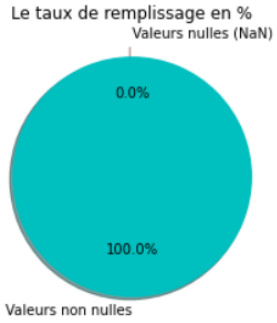
Products

```
* Nombre de colonnes sans NaN -------: 1
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 8
* Nombre de lignes ------------------------: 32951
* Nombre de colonnes ----------------------: 9
* Nombre de cases -------------------------: 296559
* Nombre de valeurs nulles ----------------: 2448
* Nombre de valeurs non nulles ------------: 294111
* le pourcentage des valeurs nulles ------: 0.8 %
* le pourcentage des valeurs non nulles --: 99.2 %
```



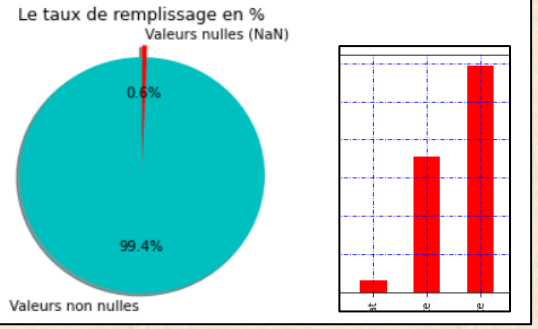
Translation

```
* Nombre de colonnes sans NaN -------: 2
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 0
* Nombre de lignes ------------------------: 71
* Nombre de colonnes ----------------------: 2
* Nombre de cases -------------------------: 142
* Nombre de valeurs nulles ----------------: 0
* Nombre de valeurs non nulles ------------: 142
* le pourcentage des valeurs nulles ------: 0.0 %
* le pourcentage des valeurs non nulles --: 100.0 %
```



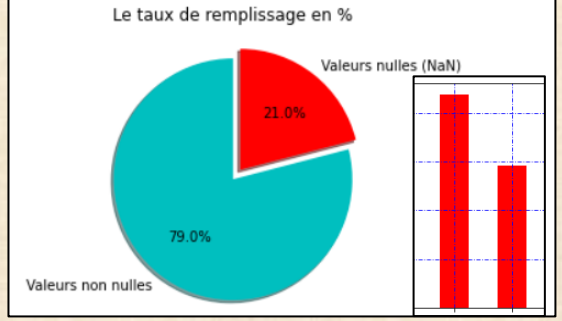
Orders

```
* Nombre de colonnes sans NaN -------: 5
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 3
* Nombre de lignes ------------------------: 99441
* Nombre de colonnes ----------------------: 8
* Nombre de cases -------------------------: 795528
* Nombre de valeurs nulles ----------------: 4908
* Nombre de valeurs non nulles ------------: 790620
* le pourcentage des valeurs nulles ------: 0.6 %
* le pourcentage des valeurs non nulles --: 99.4 %
```



Order_reviews

```
* Nombre de colonnes sans NaN -------: 5
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 2
* Nombre de lignes ------------------------: 99224
* Nombre de colonnes ----------------------: 7
* Nombre de cases -------------------------: 694568
* Nombre de valeurs nulles ----------------: 145903
* Nombre de valeurs non nulles ------------: 548665
* le pourcentage des valeurs nulles ------: 21.0 %
* le pourcentage des valeurs non nulles --: 79.0 %
```



Traitement et nettoyage du Data-Set

olist

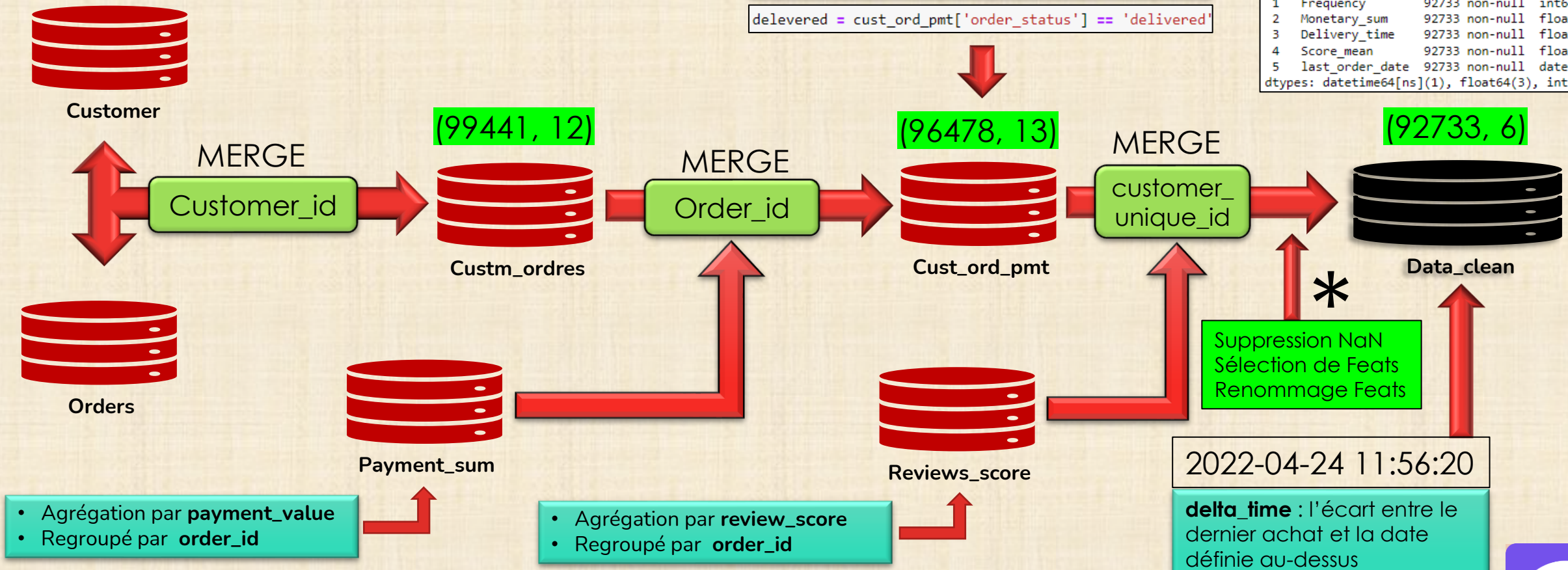
Les étapes de sélection et de création des Features:

delivery_time = delivered_customer_date - cust_ord_pmt.order_purchase_timestamp

```
delevered = cust_ord_pmt['order_status'] == 'delivered'
```

Data-Set pour Clustering

```
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Recency          92733 non-null  int64
 1   Frequency        92733 non-null  int64
 2   Monetary_sum     92733 non-null  float64
 3   Delivery_time    92733 non-null  float64
 4   Score_mean       92733 non-null  float64
 5   last_order_date  92733 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(3), int64(2)
```



- Agrégation par **payment_value**
- Regroupé par **order_id**

- Agrégation par **review_score**
- Regroupé par **order_id**

delta_time : l'écart entre le dernier achat et la date définie au-dessus

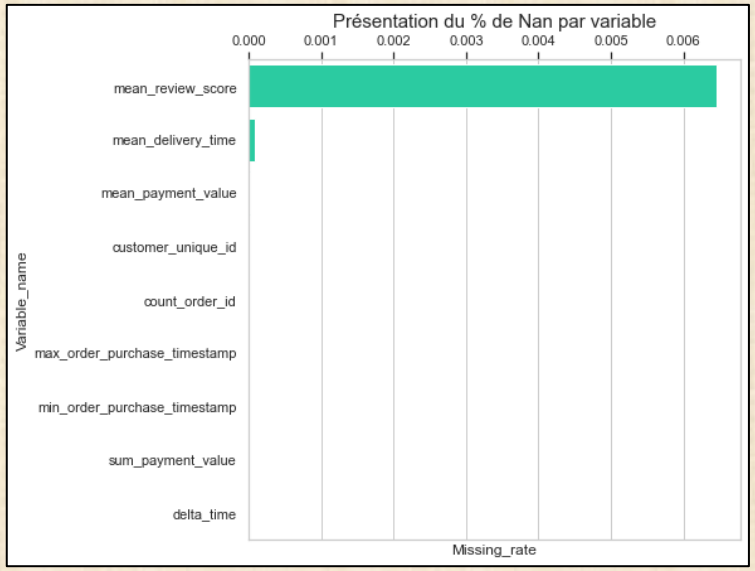
```
data['delta_time'] = (date - data['max_order_purchase_timestamp'])
```

Traitement et nettoyage du Data-Set

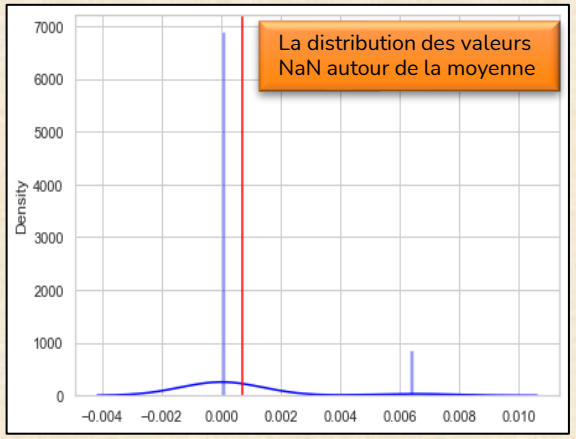
*
Suppression NaN
Sélection de Feats
Renommage Feats

olist

Suppression NaN / Sélection de Feats / Renommage Feats:



	Variable_name	Missing_values	Missing_rate
7	mean_review_score	603	0.006459
6	mean_delivery_time	8	0.000086
5	mean_payment_value	1	0.000011
0	customer_unique_id	0	0.000000
1	count_order_id	0	0.000000
2	max_order_purchase_timestamp	0	0.000000
3	min_order_purchase_timestamp	0	0.000000
4	sum_payment_value	0	0.000000
8	delta_time	0	0.000000



Suppression NaN

```
* Nombre de colonnes sans NaN -------: 9
* Nombre de colonnes NaN ------------: 0
* Nombre de colonnes mixtes----------: 0
* Nombre de ligne entièrement nulles : 0
* Nombre de ligne mixtes ----------- : 0
* Nombre de ligne sans NaN ----------: 92746
* Nombre de lignes ----------------------: 92746
* Nombre de colonnes ---------------------: 9
* Nombre de cases ------------------------: 834714
* Nombre de valeurs nulles ---------------: 0
* Nombre de valeurs non nulles -----------: 834714
* le pourcentage des valeurs nulles ------: 0.0 %
* le pourcentage des valeurs non nulles --: 100.0 %
```

Data-Set Clean

	Recency	Frequency	Monetary_sum	Delivery_time	Score_mean	last_order_date
0	1445	1	141.90	6.0	5.0	2018-05-10 10:56:27
1	1448	1	27.19	3.0	4.0	2018-05-07 11:11:27
2	1870	1	86.22	25.0	3.0	2017-03-10 21:05:03
3	1654	1	43.62	20.0	4.0	2017-10-12 20:29:41
4	1621	1	196.89	13.0	5.0	2017-11-14 19:45:42
...	...	...	...	...	...	...
93353	1780	1	2067.42	27.0	5.0	2017-06-08 21:00:36
93354	1595	1	84.58	30.0	4.0	2017-12-10 20:07:56
93355	1901	1	112.46	14.0	5.0	2017-02-07 15:49:16
93356	1452	1	133.69	11.0	5.0	2018-05-02 15:17:41
93357	1817	1	71.56	7.0	5.0	2017-05-02 20:18:45

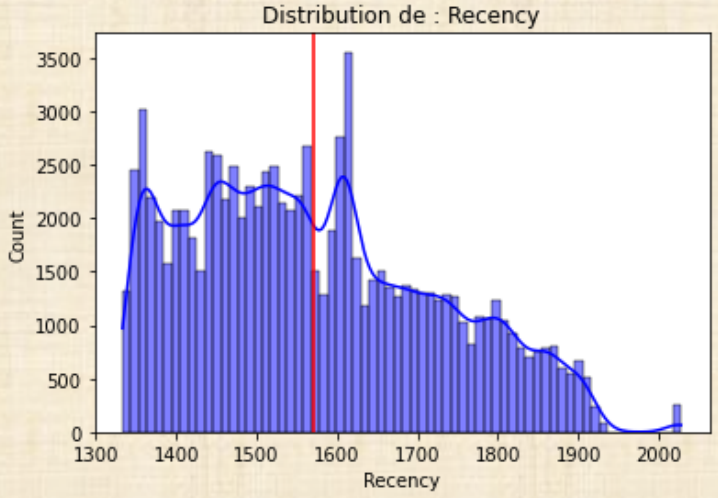
Sélection de Feats

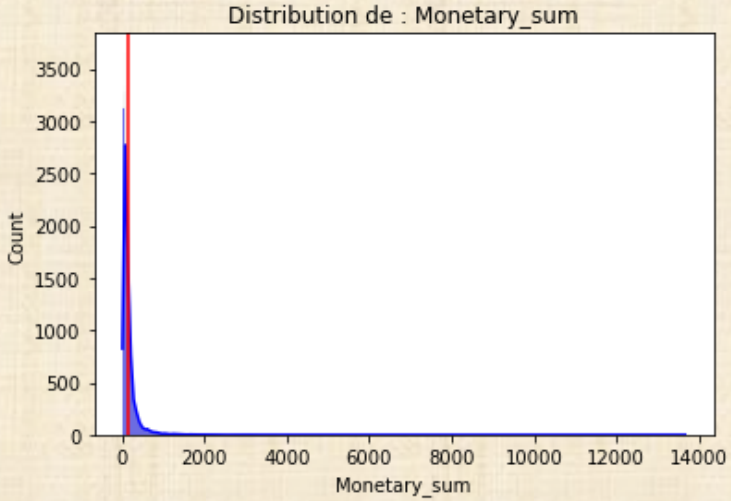
```
{'customer_unique_id'           : 'unique_id',
 'count_order_id'               : 'Frequency',
 'max_order_purchase_timestamp': 'last_order_date',
 'min_order_purchase_timestamp': 'min_order_date',
 'sum_payment_value'            : 'Monetary_sum',
 'mean_payment_value'           : 'Monetary_mean',
 'mean_delivery_time'           : 'Delivery_time',
 'mean_review_score'            : 'Score_mean',
 'delta_time'                   : 'Recency'}
```

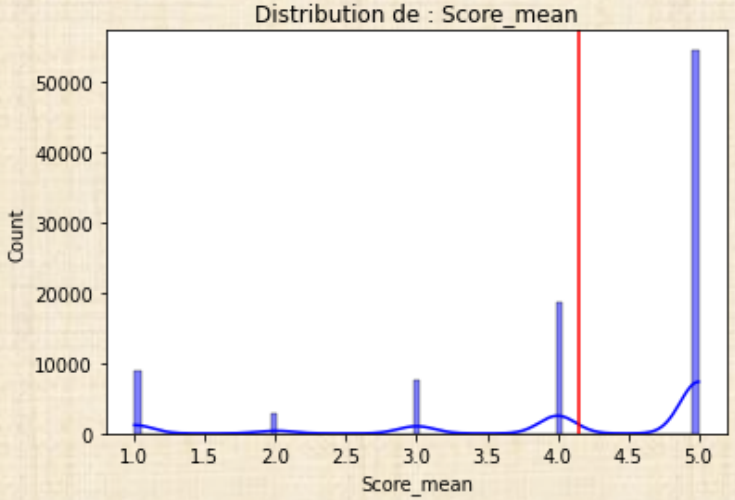
Renommage Feats

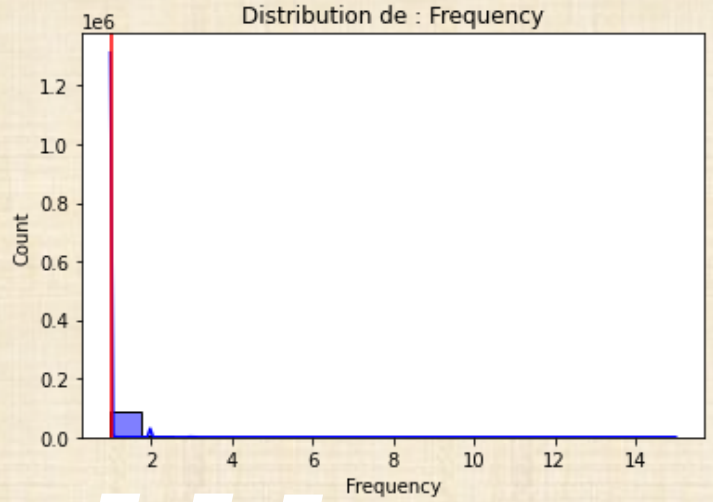
Analyse exploratoire

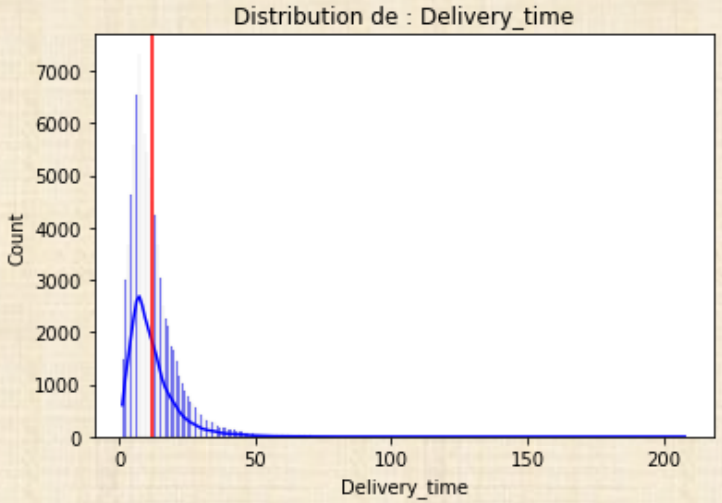
Analyse univariée: Distribution des données quantitatives par rapport à la moyenne (**Data-Set préparé**)

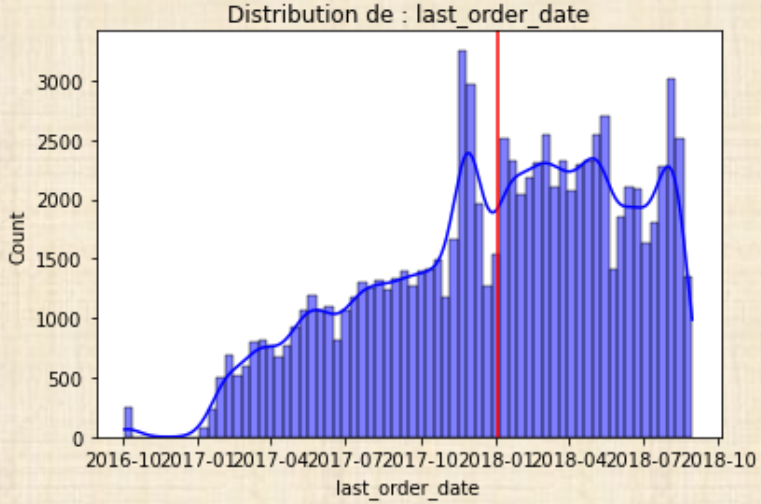








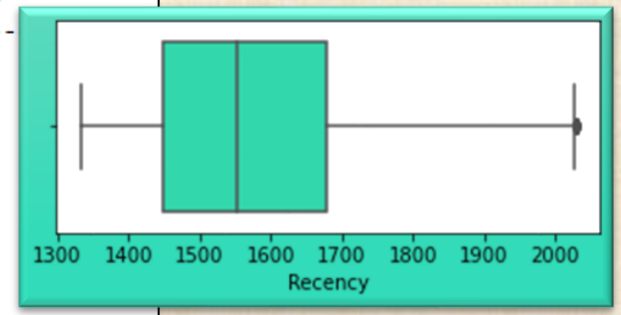


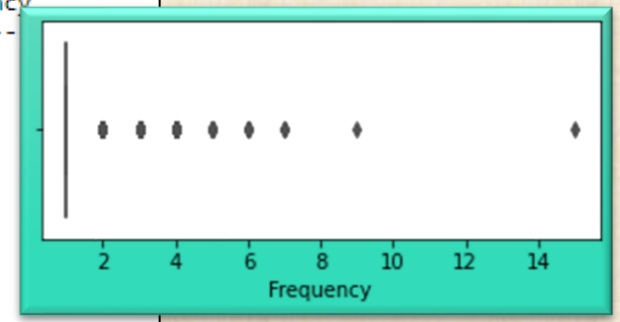


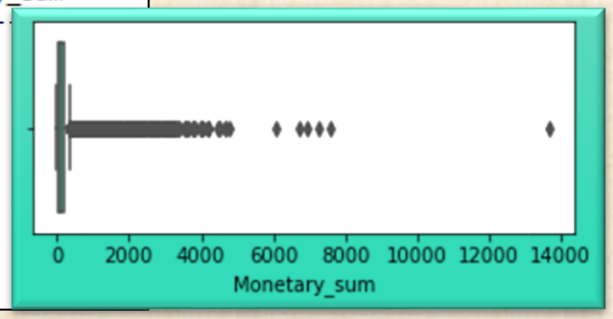
Analyse exploratoire

Analyse univariée: distribution quantitatives et qualitatives

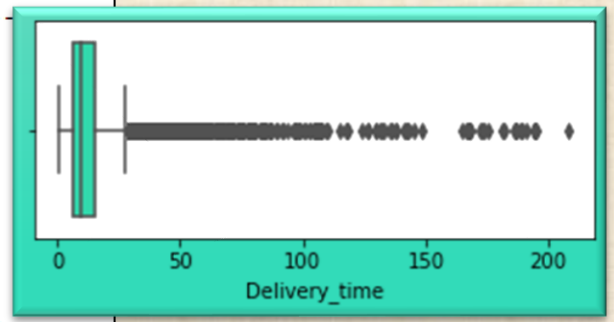
```
Indicateurs de distribution pour Recency
----------------------------------------
count    92733.000000
mean      1570.578661
std        152.586071
min       1333.000000
25%       1447.000000
50%       1552.000000
75%       1679.000000
max       2028.000000
Name: Recency, dtype: float64
Indicateurs de distribution pour Frequency
----------------------------------------
count    92733.000000
mean         1.033548
std          0.209567
min          1.000000
25%          1.000000
50%          1.000000
75%          1.000000
max         15.000000
Name: Frequency, dtype: float64
Indicateurs de distribution pour Monetary_sum
----------------------------------------
count    92733.000000
mean       164.953753
std        225.061569
min          9.590000
25%         63.010000
50%        107.780000
75%        182.320000
max      13664.080000
Name: Monetary_sum, dtype: float64
```

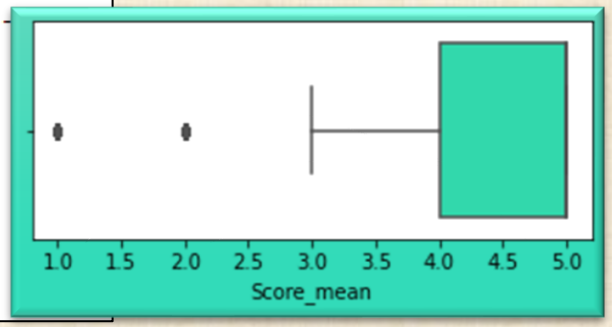







```
Indicateurs de distribution pour Delivery_time
----------------------------------------
count    92733.000000
mean        12.063451
std          9.465223
min          1.000000
25%          6.000000
50%         10.000000
75%         15.000000
max        208.000000
Name: Delivery_time, dtype: float64
Indicateurs de distribution pour Score_mean
----------------------------------------
count    92733.000000
mean         4.152589
std          1.280323
min          1.000000
25%          4.000000
50%          5.000000
75%          5.000000
max          5.000000
Name: Score_mean, dtype: float64
```

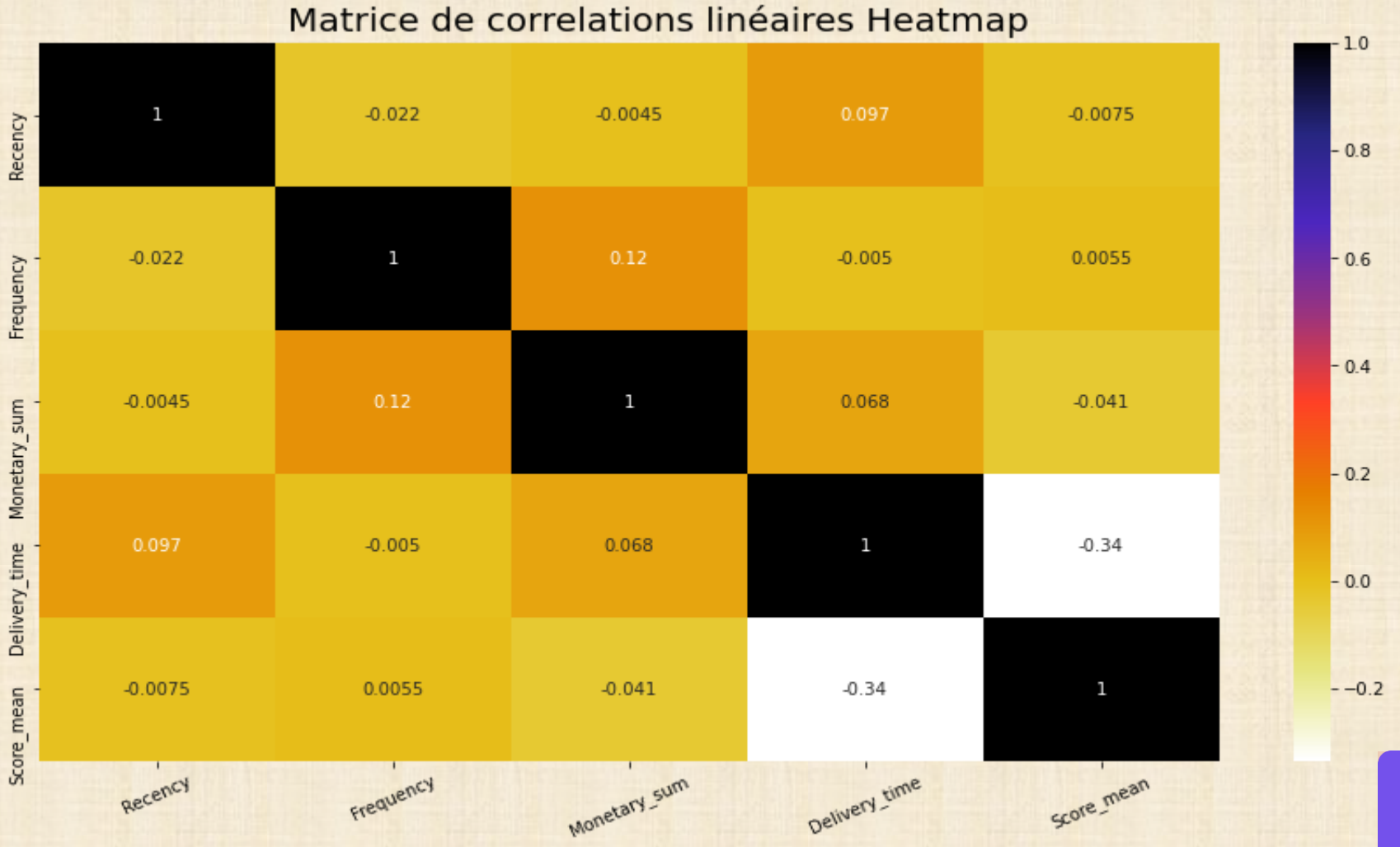




C. Analyse exploratoire

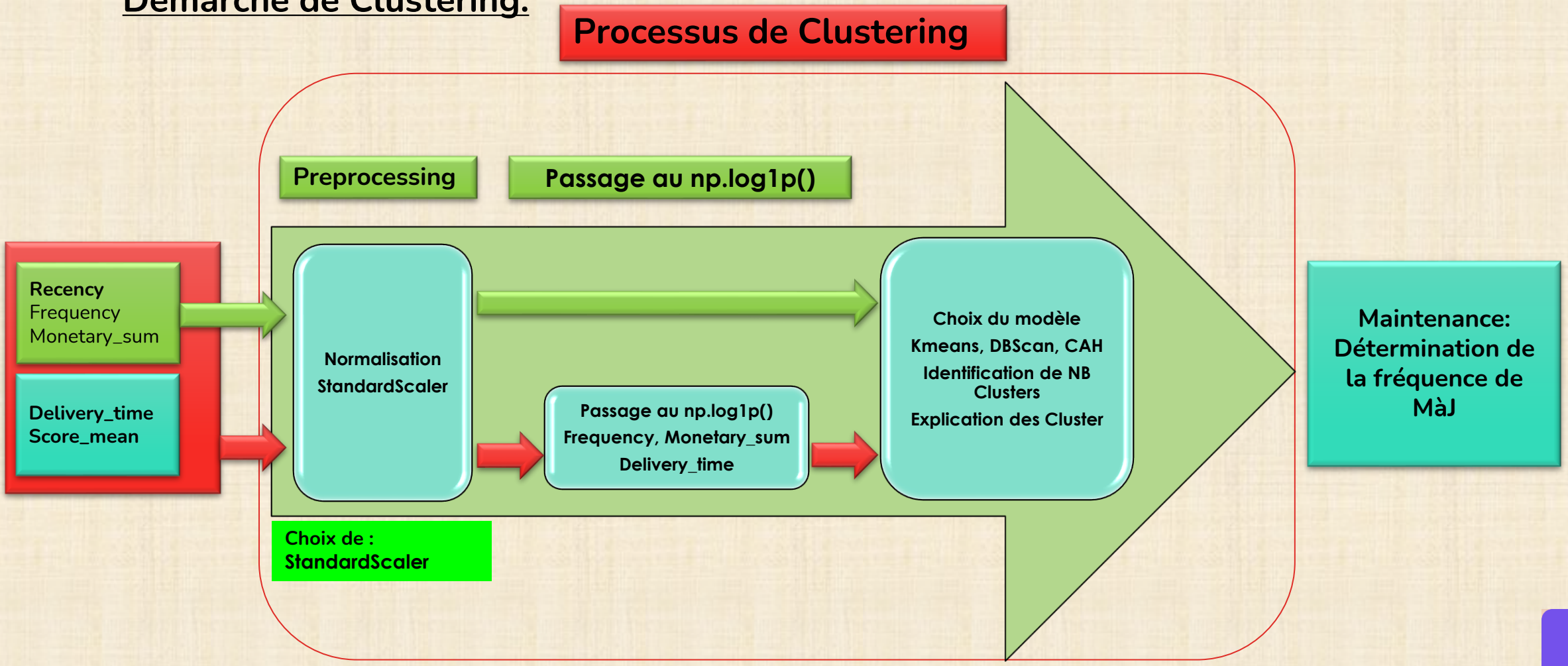
Analyse bivariée:

- Comme illustré au-dessus on voit bien qu'il n y a pas de corrélation entre nos Features, qui est bien pour notre la classification non supervisée « Clustering »



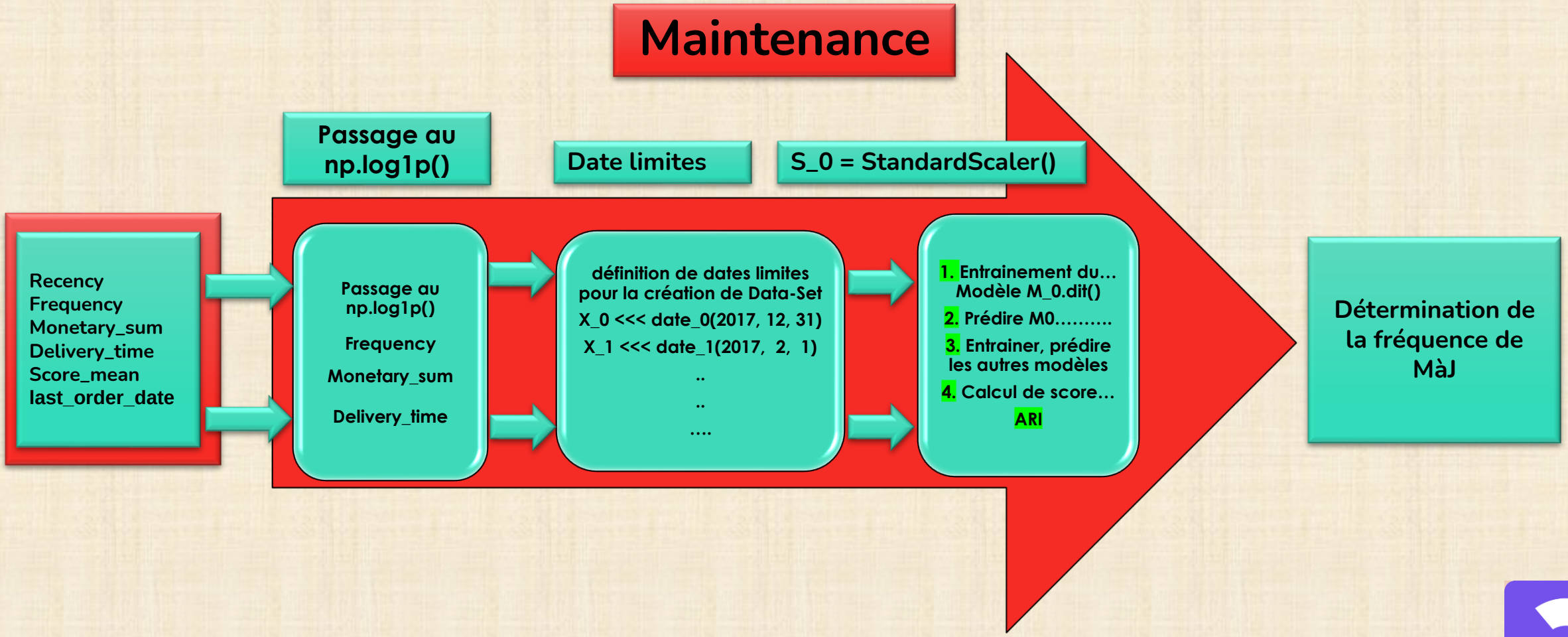
Modélisation

Démarche de Clustering:



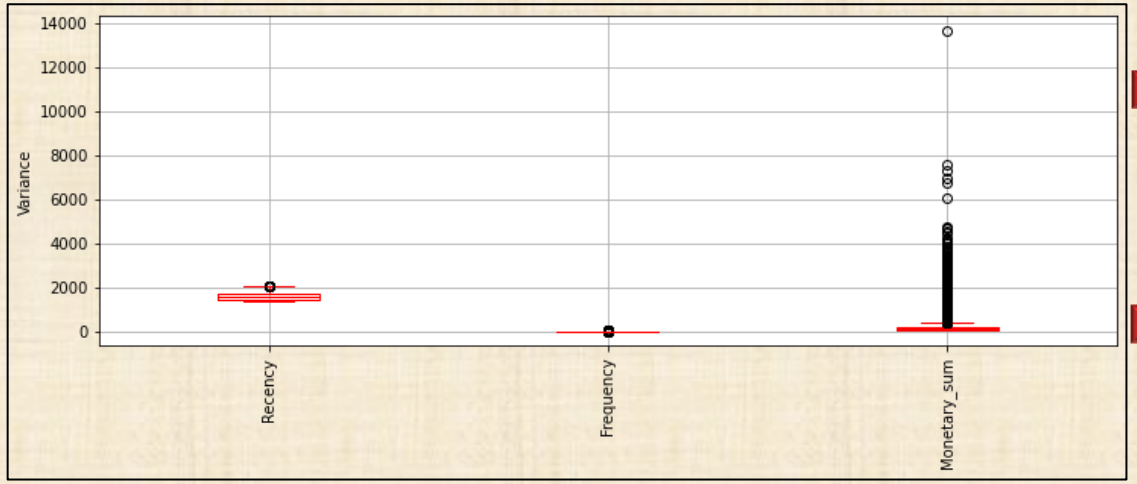
Modélisation

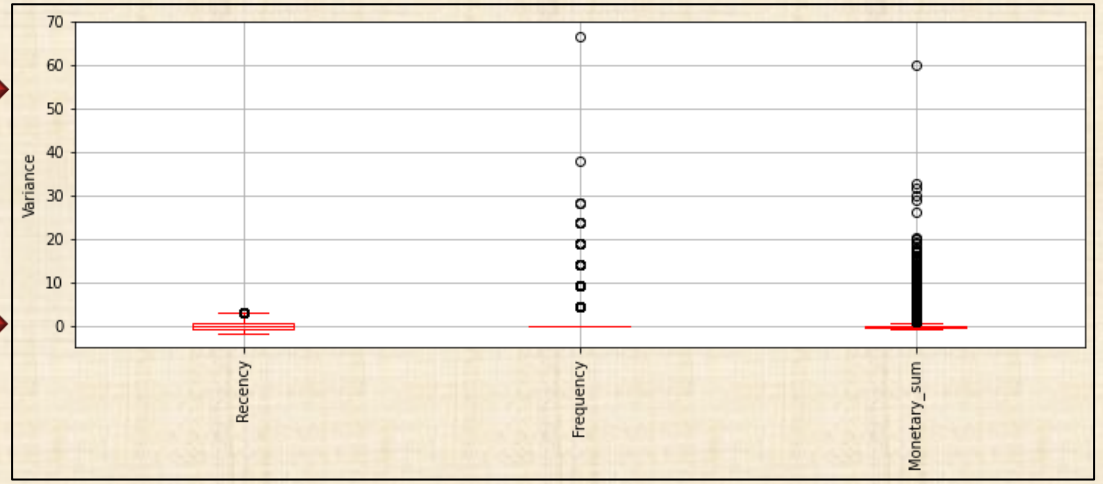
Démarche de maintenance:

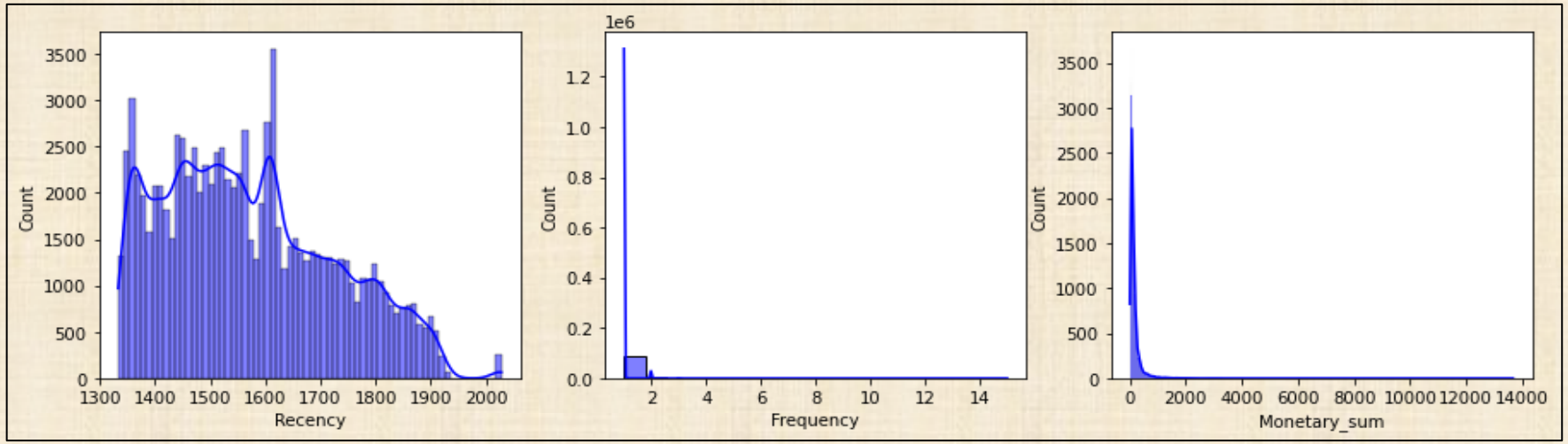


Modélisation 1

Preprocessing : Recency / Frequency / Monetary_sum sans passage au Log







* Recency: en dessous de la moyenne

* Frequency : très faible variance

* Monetary: forte variance

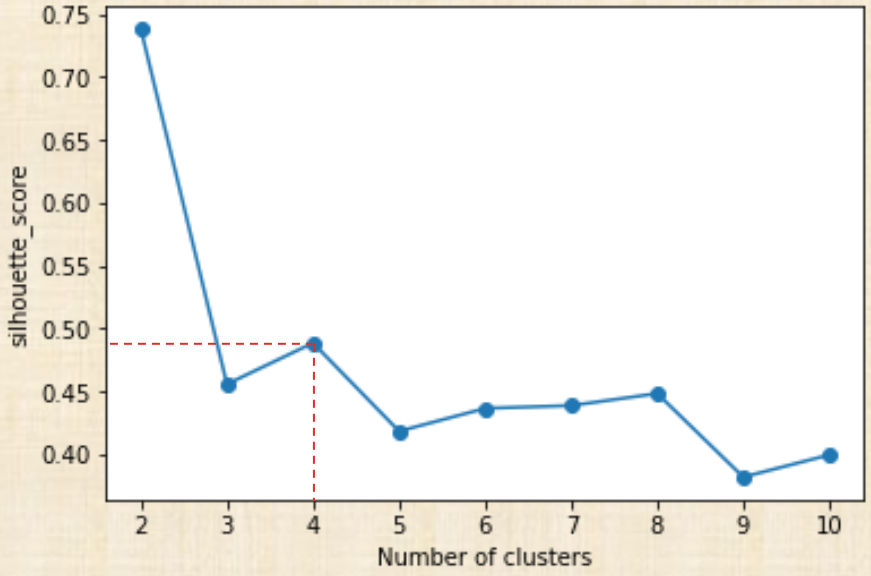
Modélisation 1

Test du modèle Kmeans

Elbow Methode: pour détecter la zone "coude" dans la minimisation du coût inertia_ afin de déterminer le nombre de Cluster K

Silhouette score

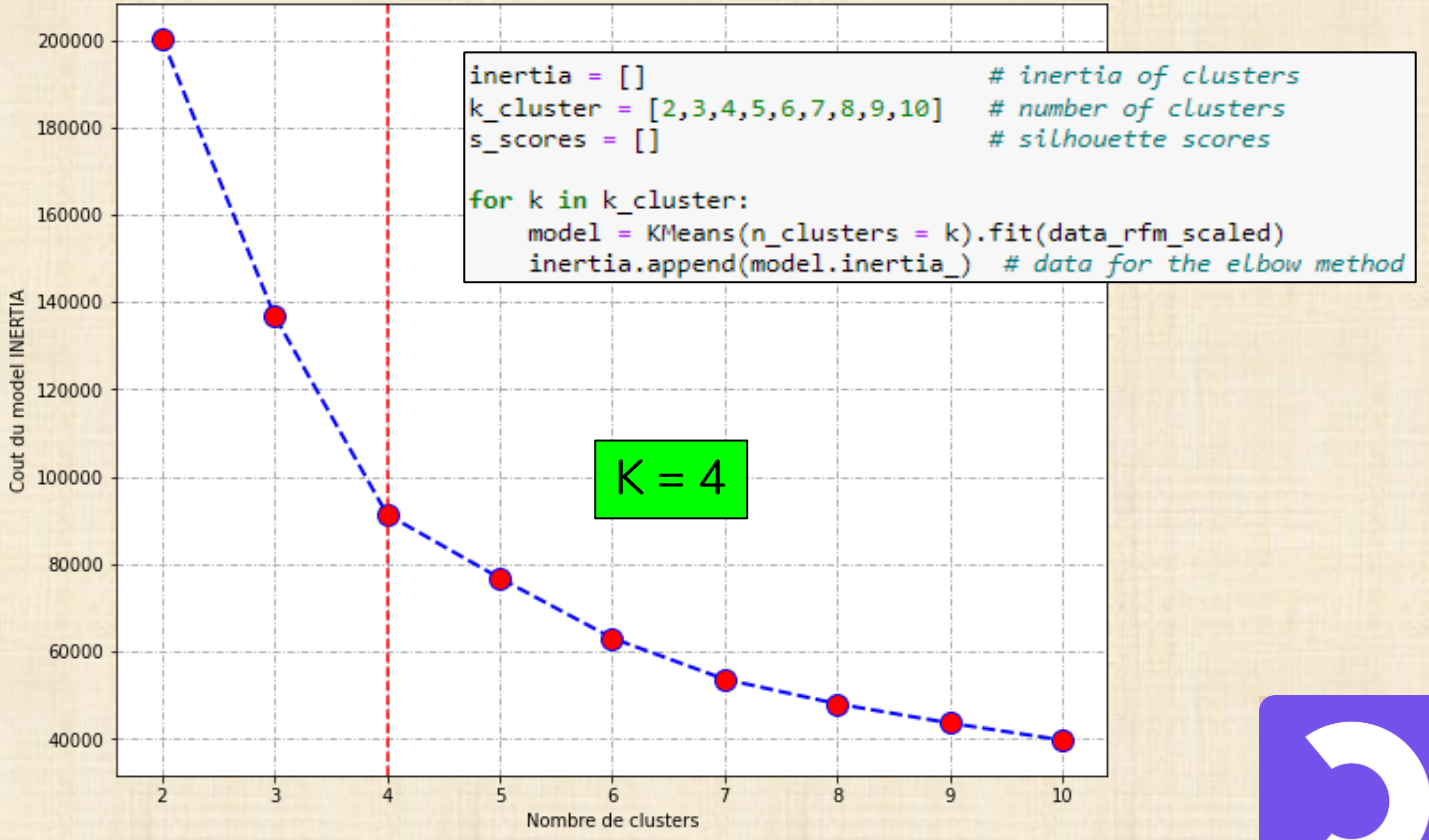
```
silhouette_avg = silhouette_score(data_rfm_scaled, model.labels_)
s_scores.append(silhouette_avg) # data for the silhouette score m
```



Elbow Methode

```
inertia = []                                # inertia of clusters
k_cluster = [2,3,4,5,6,7,8,9,10]            # number of clusters
s_scores = []                               # silhouette scores

for k in k_cluster:
    model = KMeans(n_clusters = k).fit(data_rfm_scaled)
    inertia.append(model.inertia_)  # data for the elbow method
```



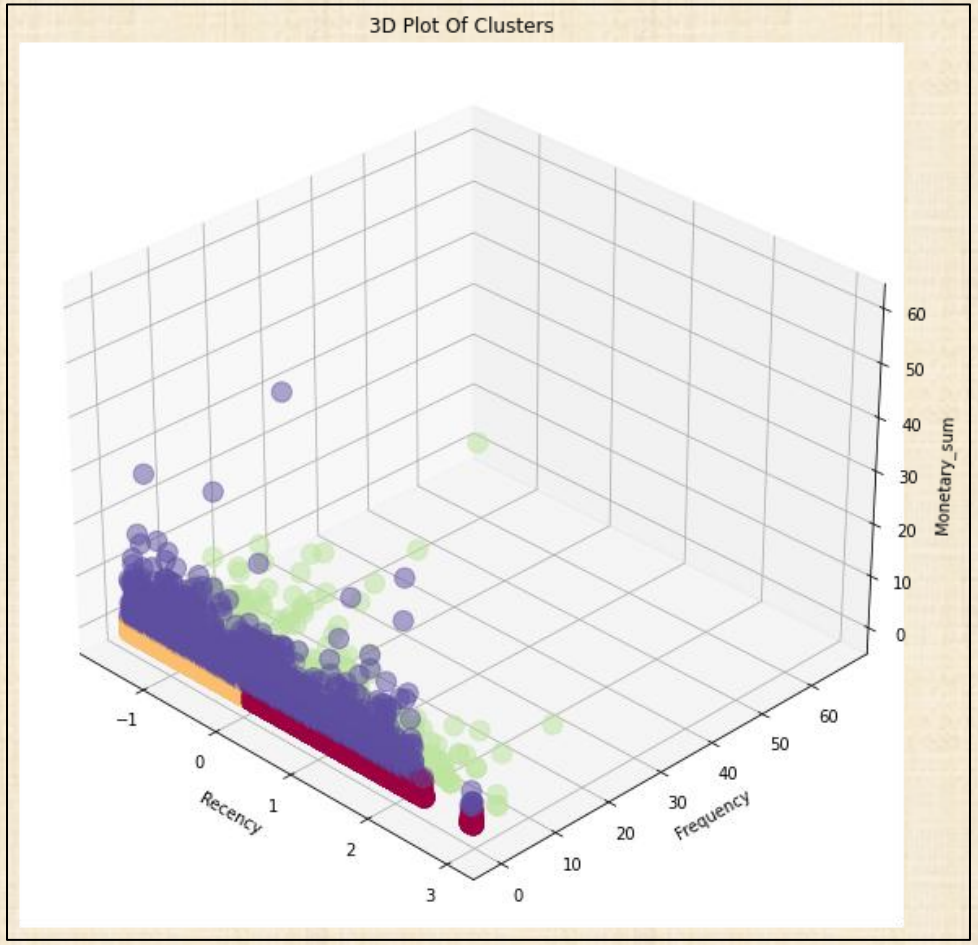
Modélisation 1

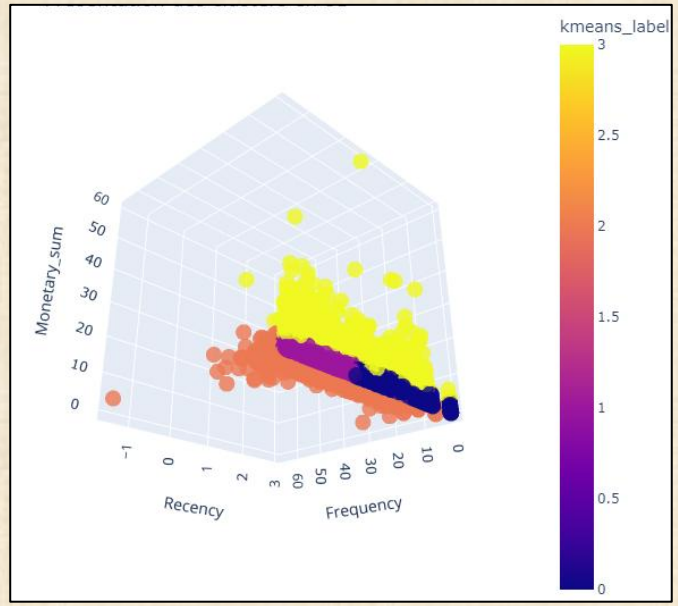
Clustering avec l'algorithme du Kmeans: K = 4

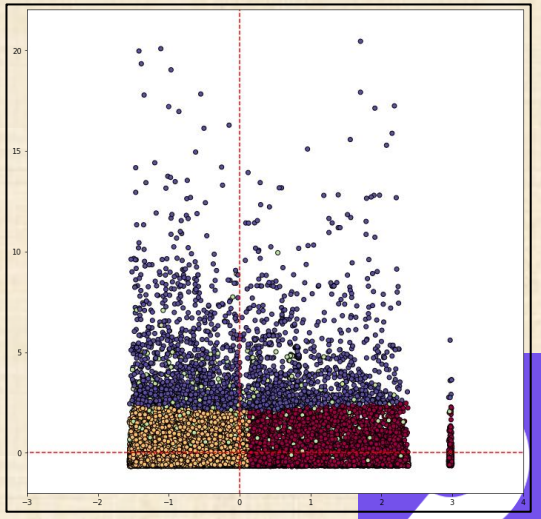
```
kmeans_model = KMeans(n_clusters = 4)
kmeans_model.fit(data_rfm_scaled)

data_rfm_scaled['kmeans_label'] = kmeans_model.labels_
data_rfm_scaled
```

	Recency	Frequency	Monetary_sum	kmeans_label
0	-0.823007	-0.160083	-0.102434	1
1	-0.803345	-0.160083	-0.612119	1
2	1.962322	-0.160083	-0.349834	0
3	0.546720	-0.160083	-0.539117	0
4	0.330447	-0.160083	0.141901	0
...	...	...	...	...
92728	1.372487	-0.160083	8.453138	3



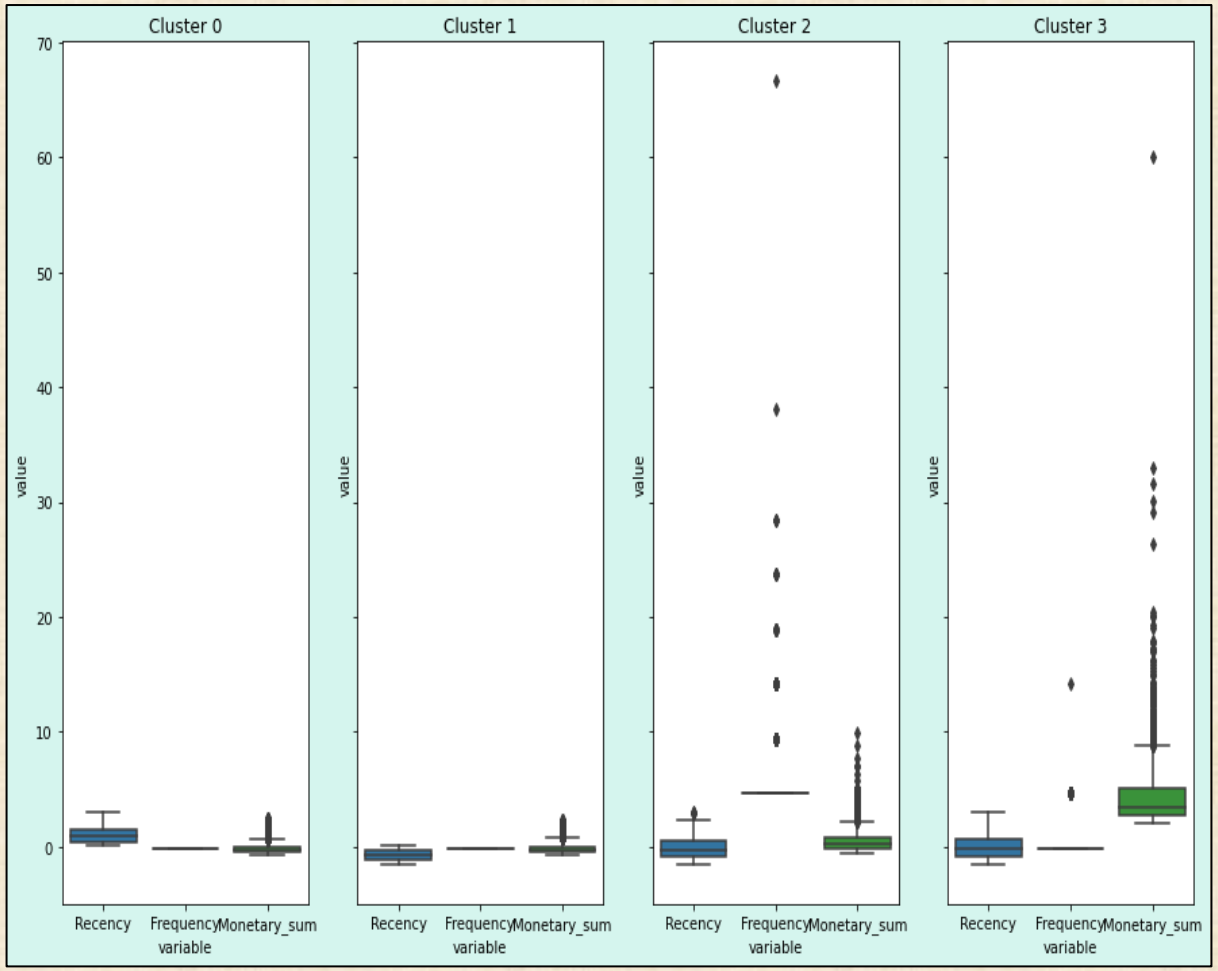




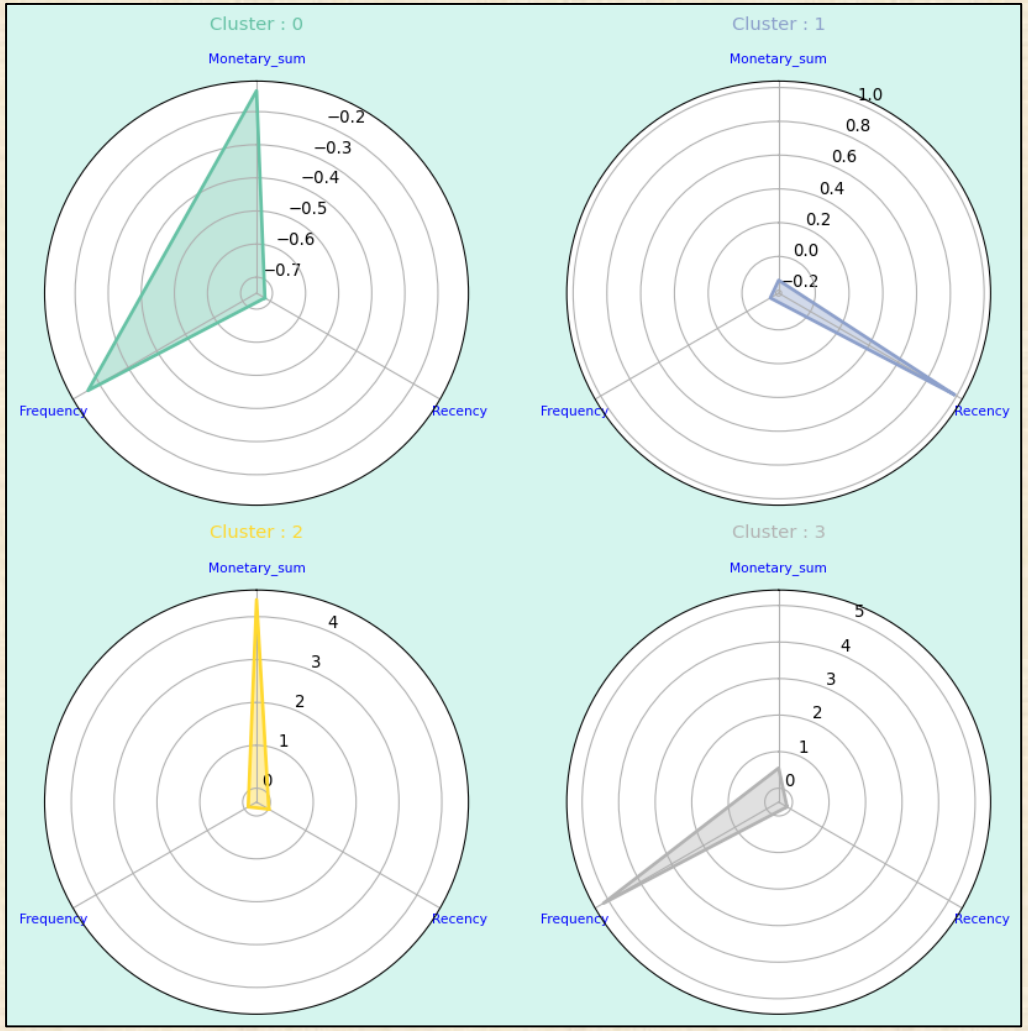
Modélisation 1

Présentation des Clusters

Box Plot



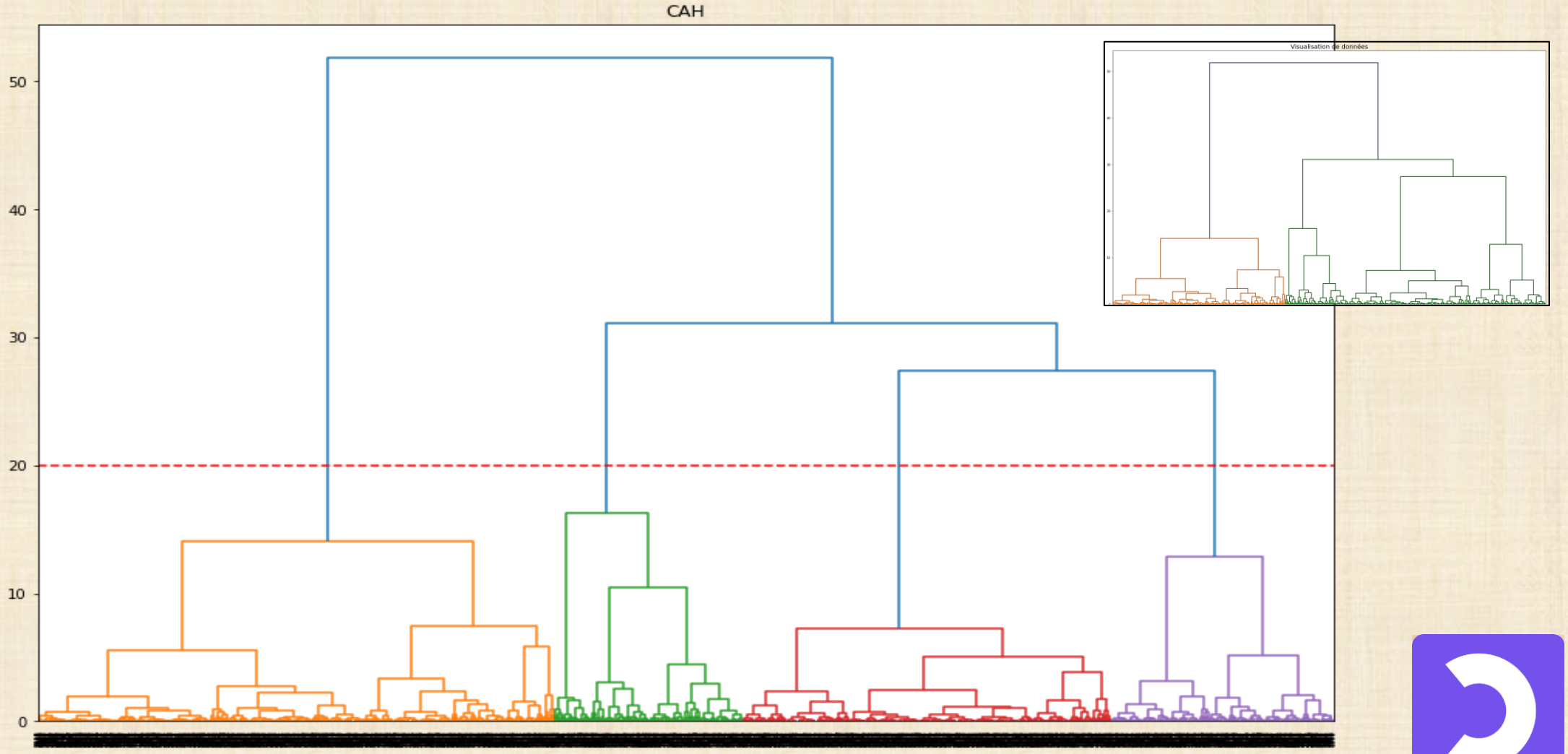
Radar Plot



Modélisation 1

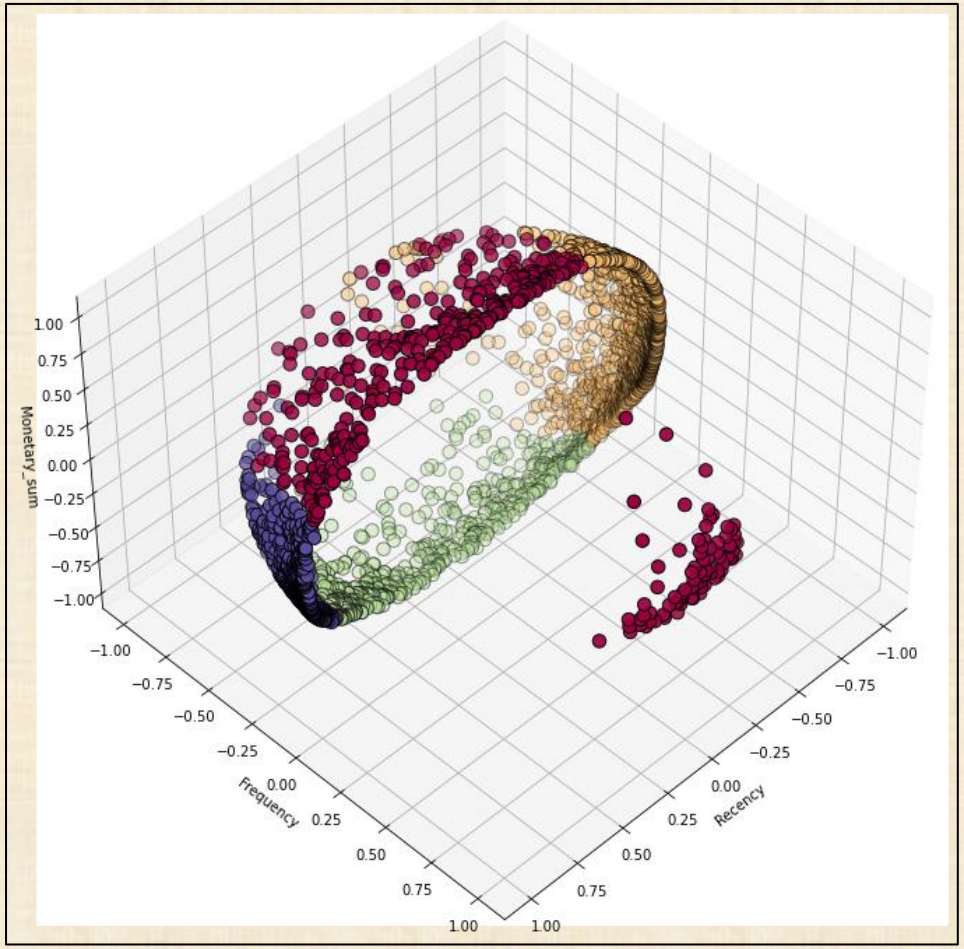
Classification ascendante hiérarchique CAH: Appliquée sur un échantillon de 2% du Data-Sets

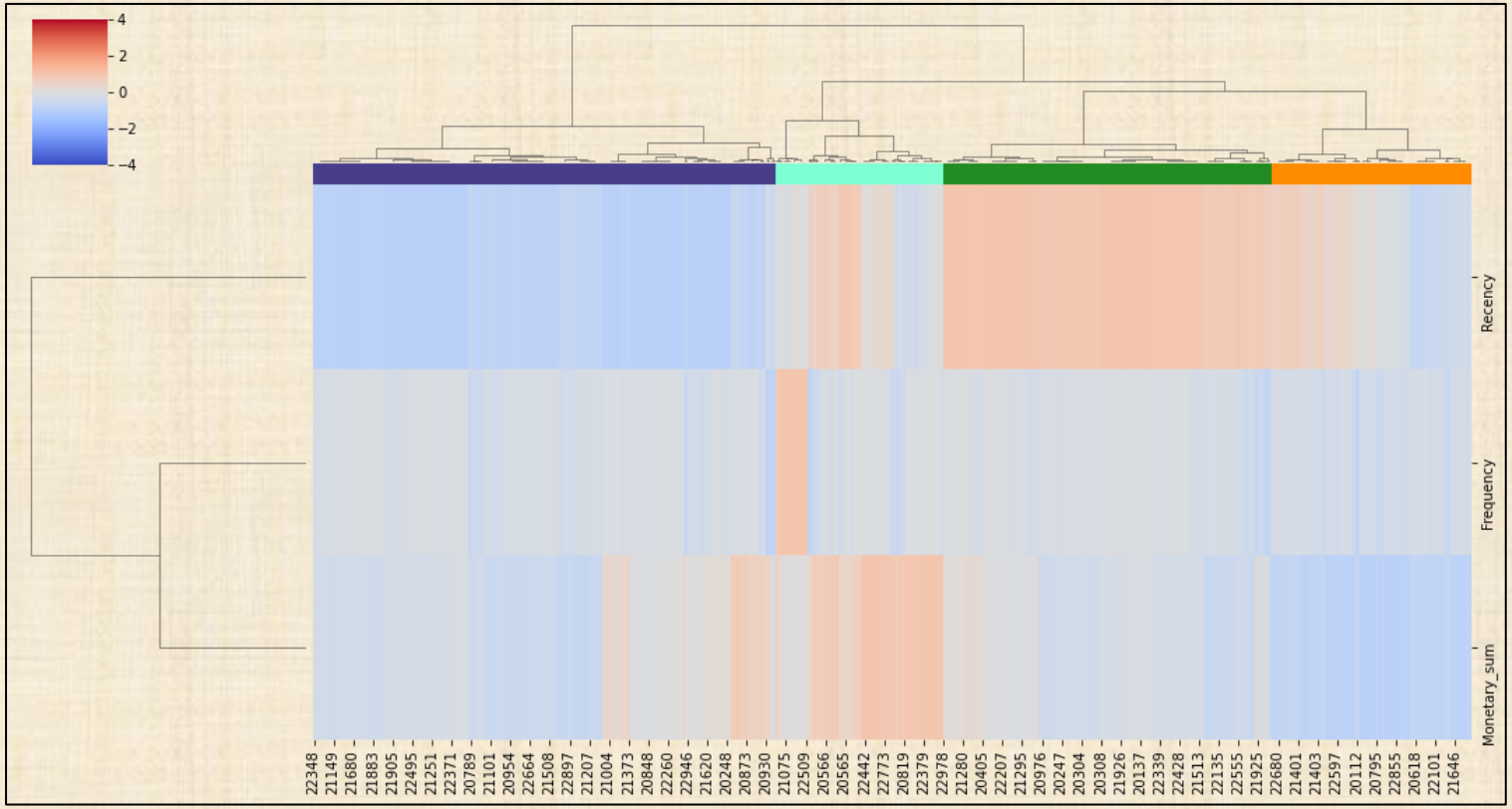
- A l'aide des scores de silhouette, on pourra définir le nombre optimal de clusters pour les données et la technique de clustering



Modélisation 1

Classification ascendante hiérarchique CAH: Appliquée sur un échantillon de 2% du Data-Sets

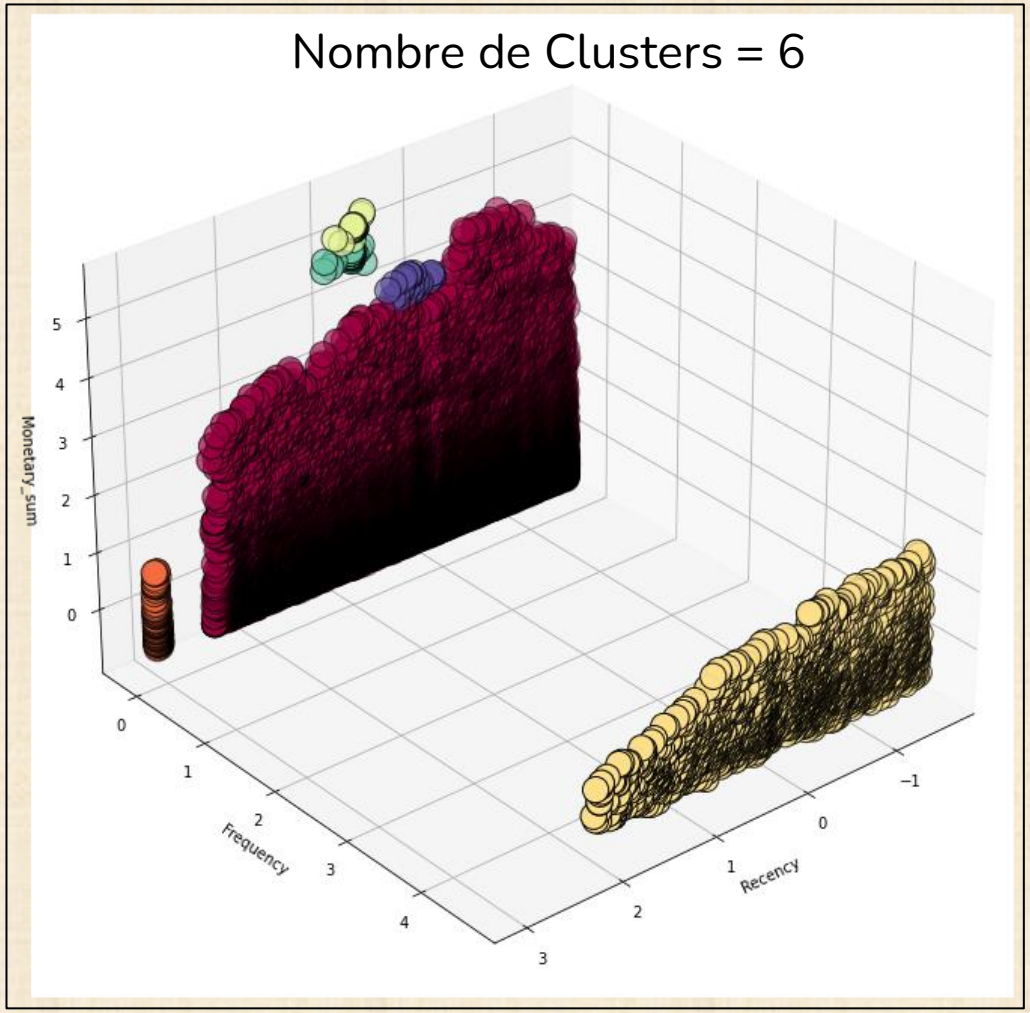




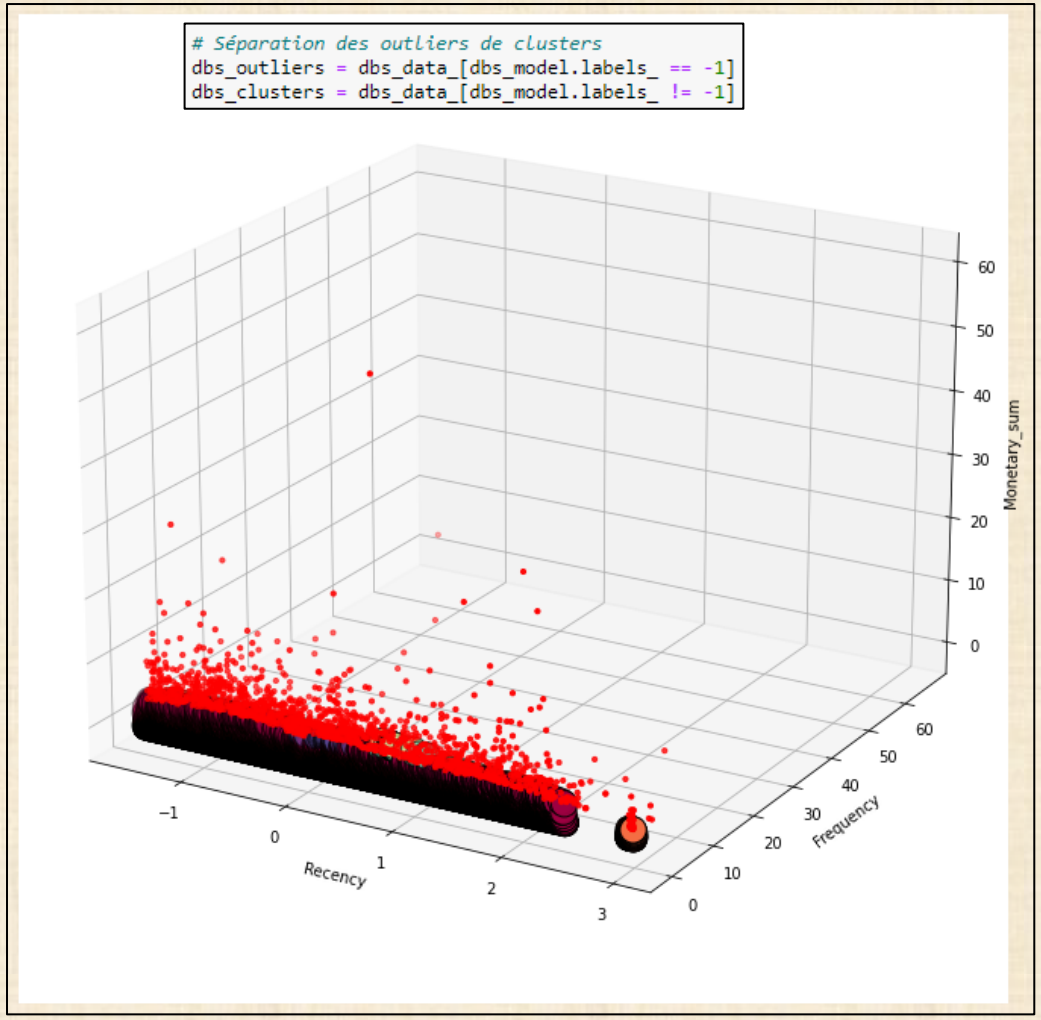
Normalisation des données afin que les données suivent approximativement une distribution gaussienne
cah_data_normalized = normalize(X_)

Modélisation 1

Classification par DBSCAN:

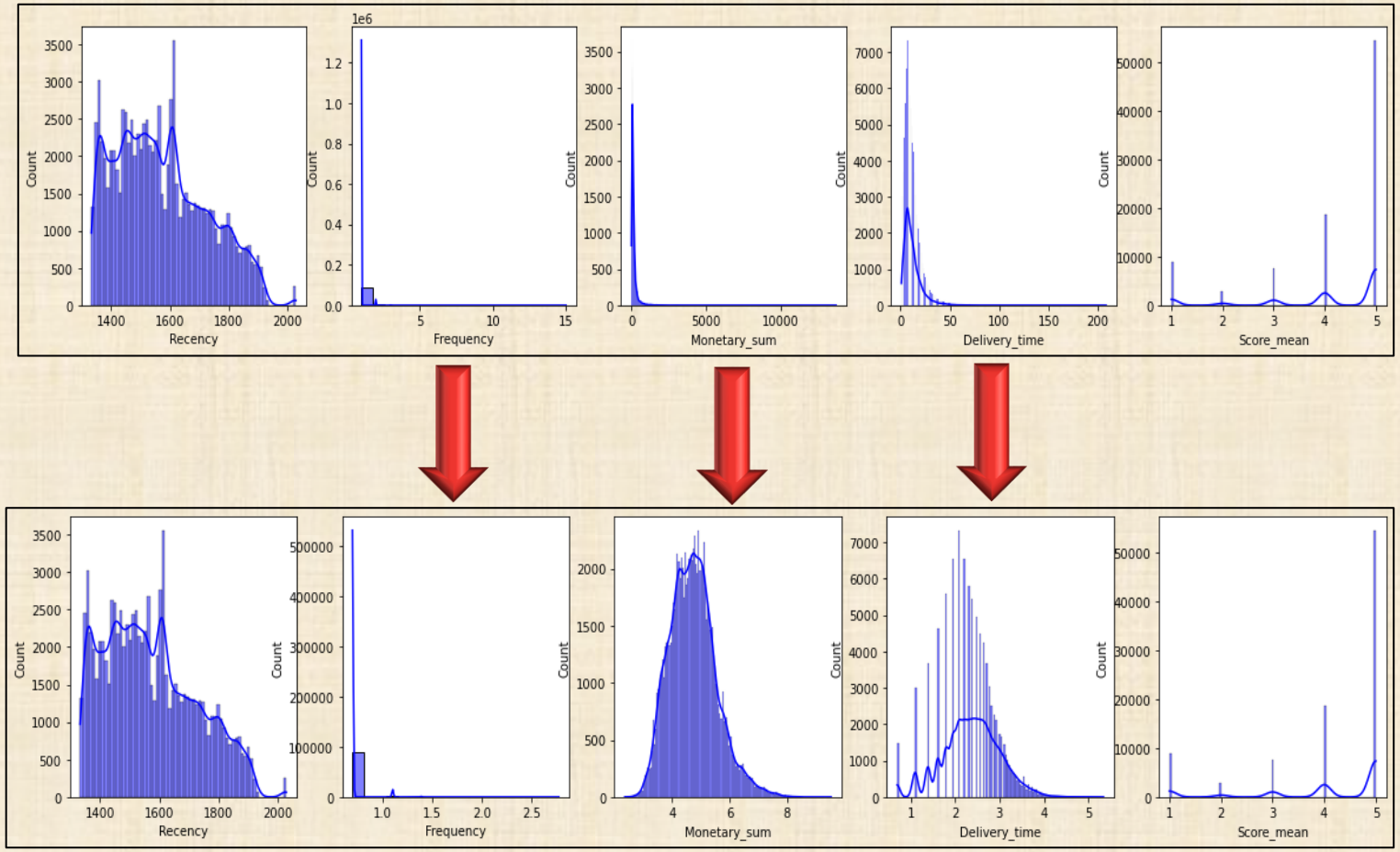


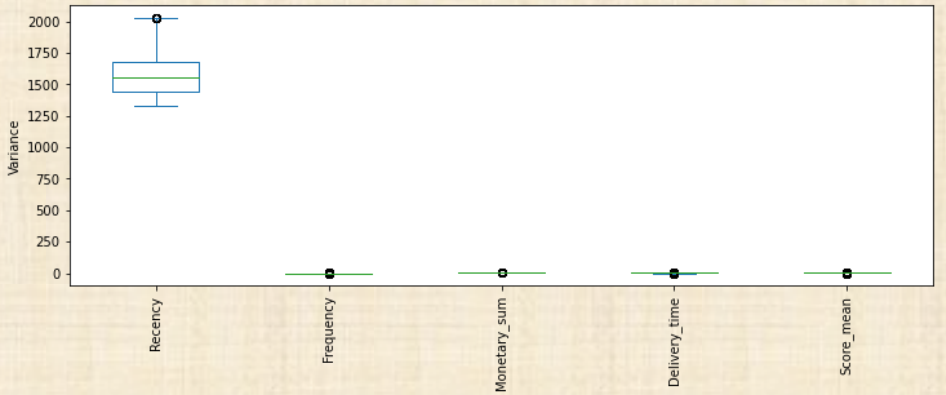

```
# Séparation des outliers de clusters
dbs_outliers = dbs_data_[dbs_model.labels_ == -1]
dbs_clusters = dbs_data_[dbs_model.labels_ != -1]
```

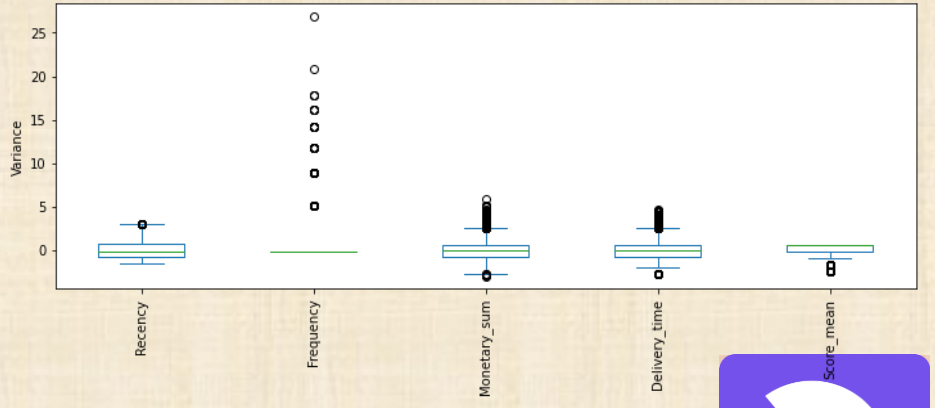


Modélisation 2

Preprocessing : passage au Log de Frequency, Monetary_sum et Delivery_time



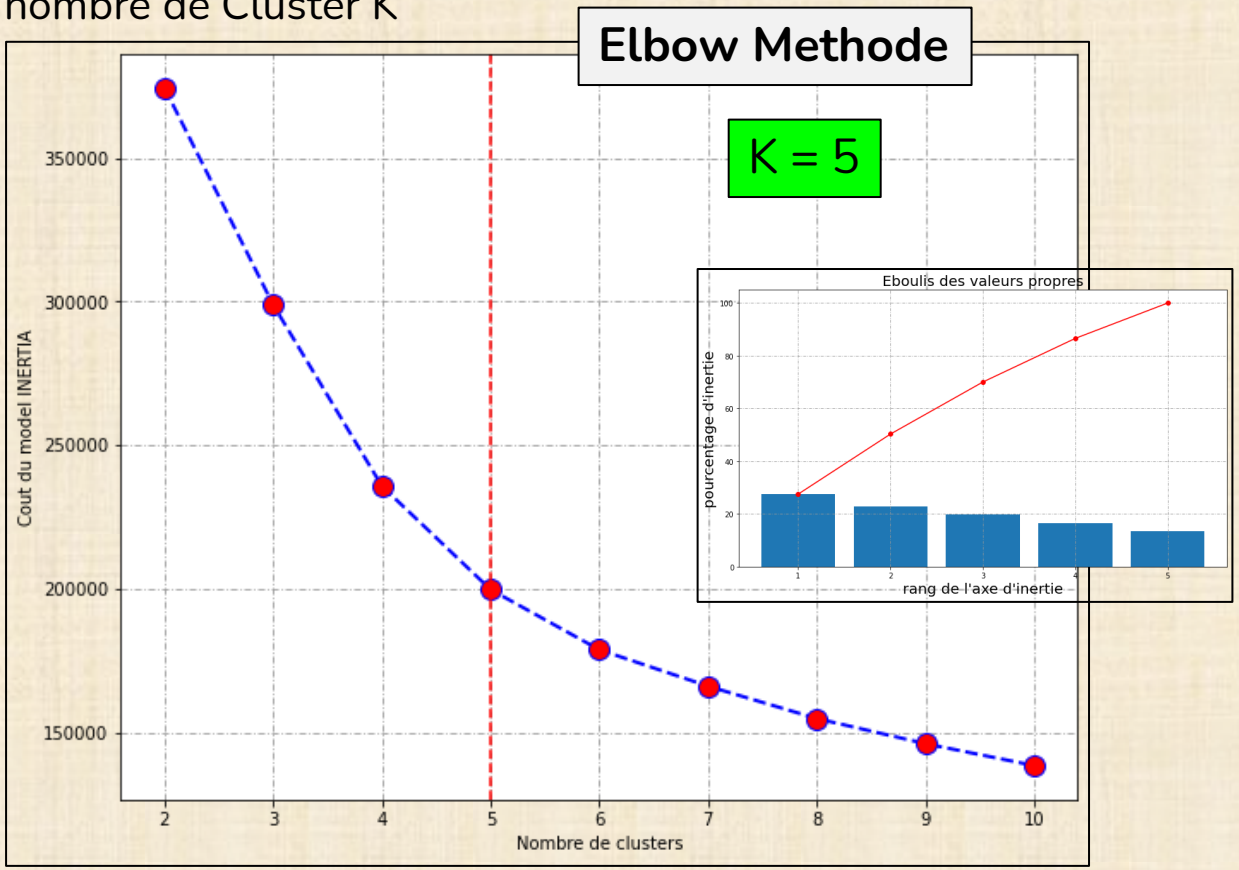


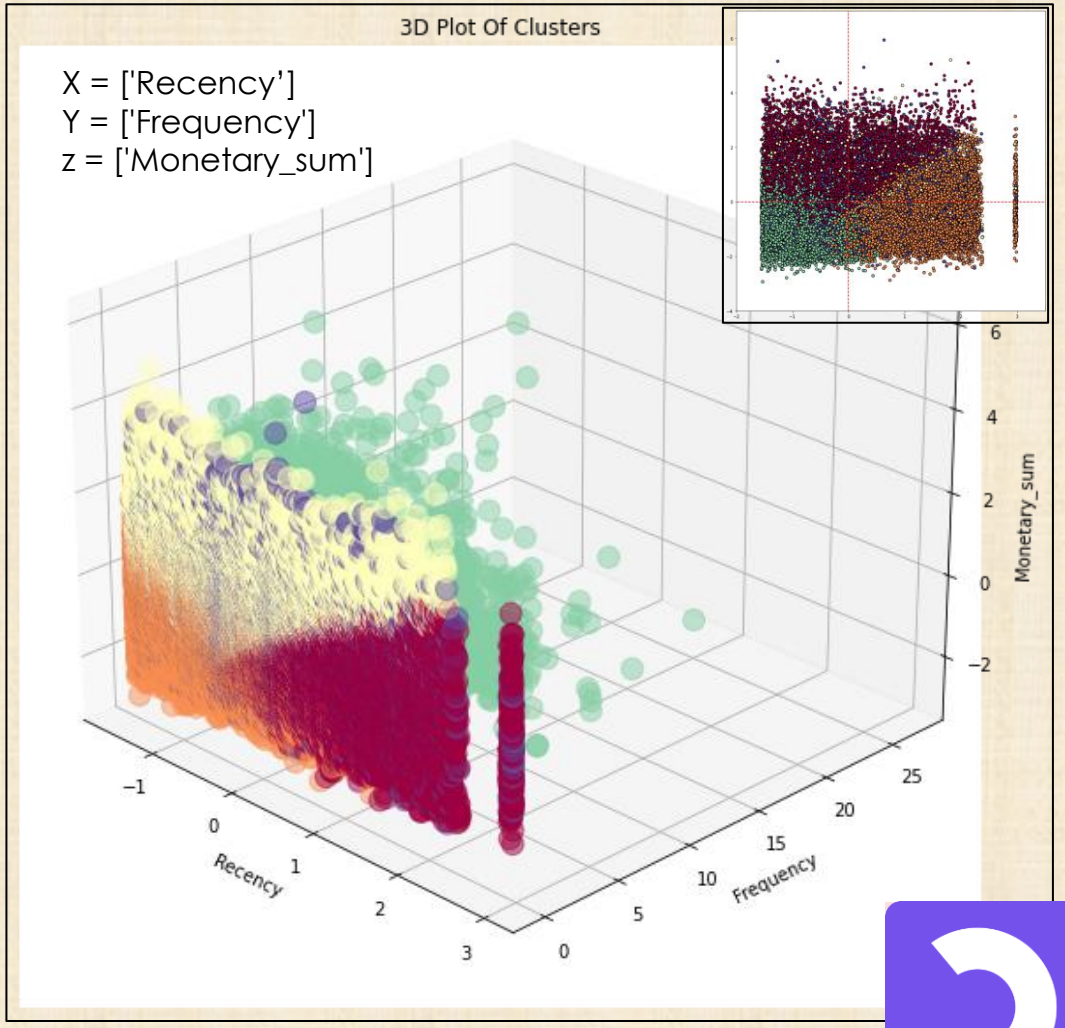


Modélisation 2

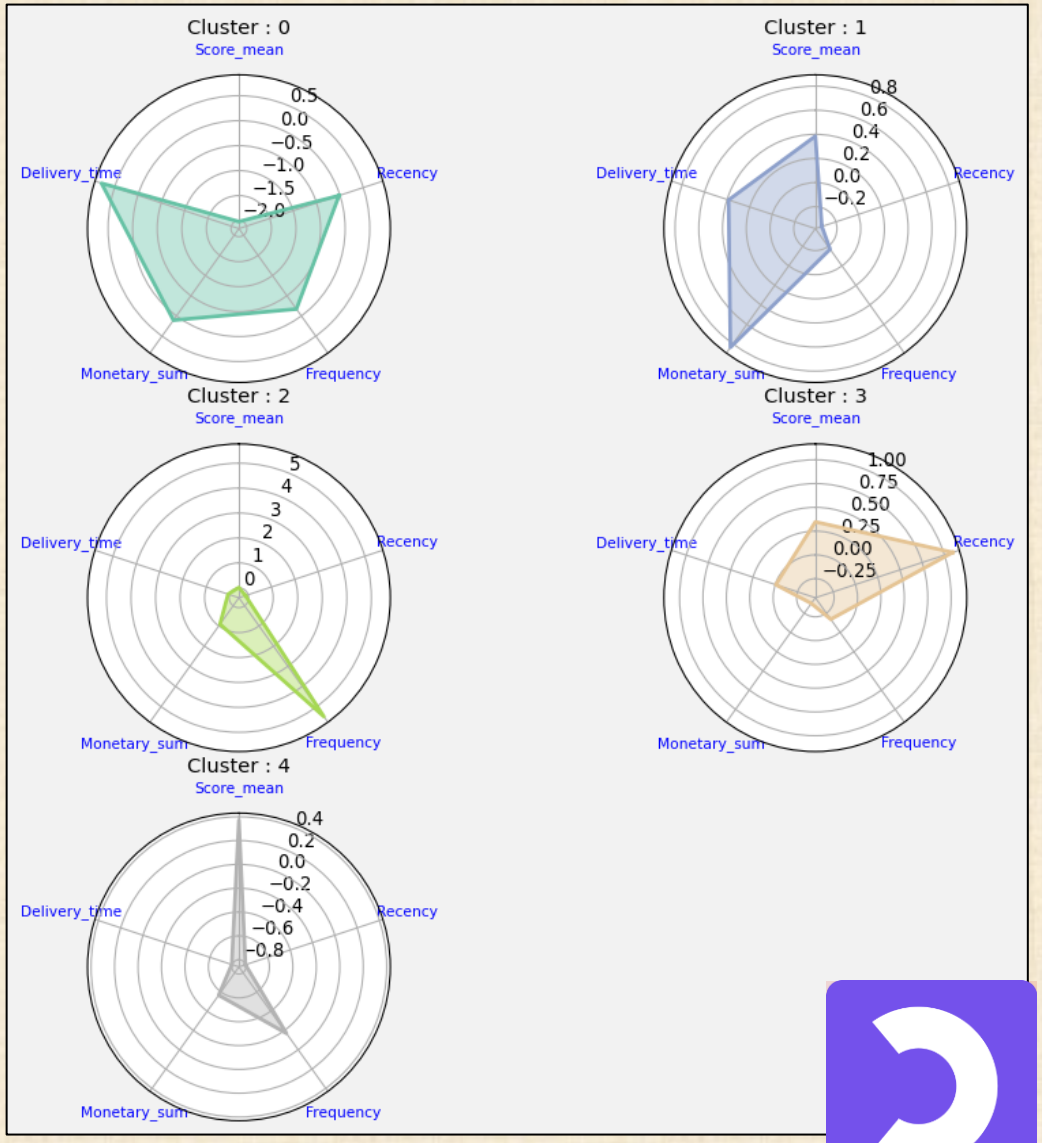
Test du modèle Kmeans

Elbow Methode: pour détecter la zone "coude" dans la minimisation du coût inertia_ afin de déterminer le nombre de Cluster K



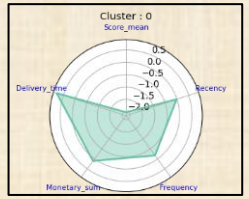


Modélisation 2

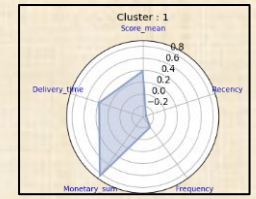


Modélisation 2

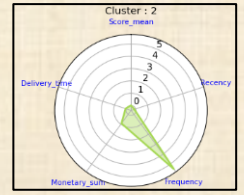
Interprétation des Clusters :



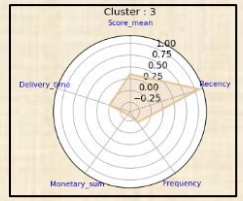
Clusters 0 : des clients avec de longs délais de livraison, au niveau de leurs avis sont mécontents « mauvais avis », avec un nombre de commandes assez élevé au dessus de la moyenne. Des anciens commandes passées, avec des montants dépensés assez élevés.



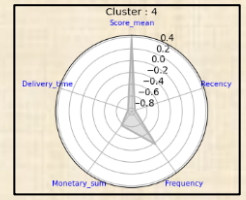
Clusters 1 : des clients avec un moyen délais de livraison, au niveau de leurs avis sont assez contents « moyenne note d'avis », avec un nombre de commandes faible. Des commandes passées récemment, avec des montants dépensés très élevés(peut être des articles assez chères).



Clusters 2 : des clients avec très faible délais de livraison, au niveau de leurs avis sont mécontents « faible note d'avis », avec un nombre de commandes très élevés. Ce sont des commandes passées récemment, avec des montants dépensés faible.



Clusters 3 : des clients avec délais de livraison proche du moyen, au niveau de leurs avis sont satisfaits « moyenne note d'avis », avec un nombre de commandes faible. Ce sont des commandes passées depuis longtemps, avec des montants dépensés très faible.

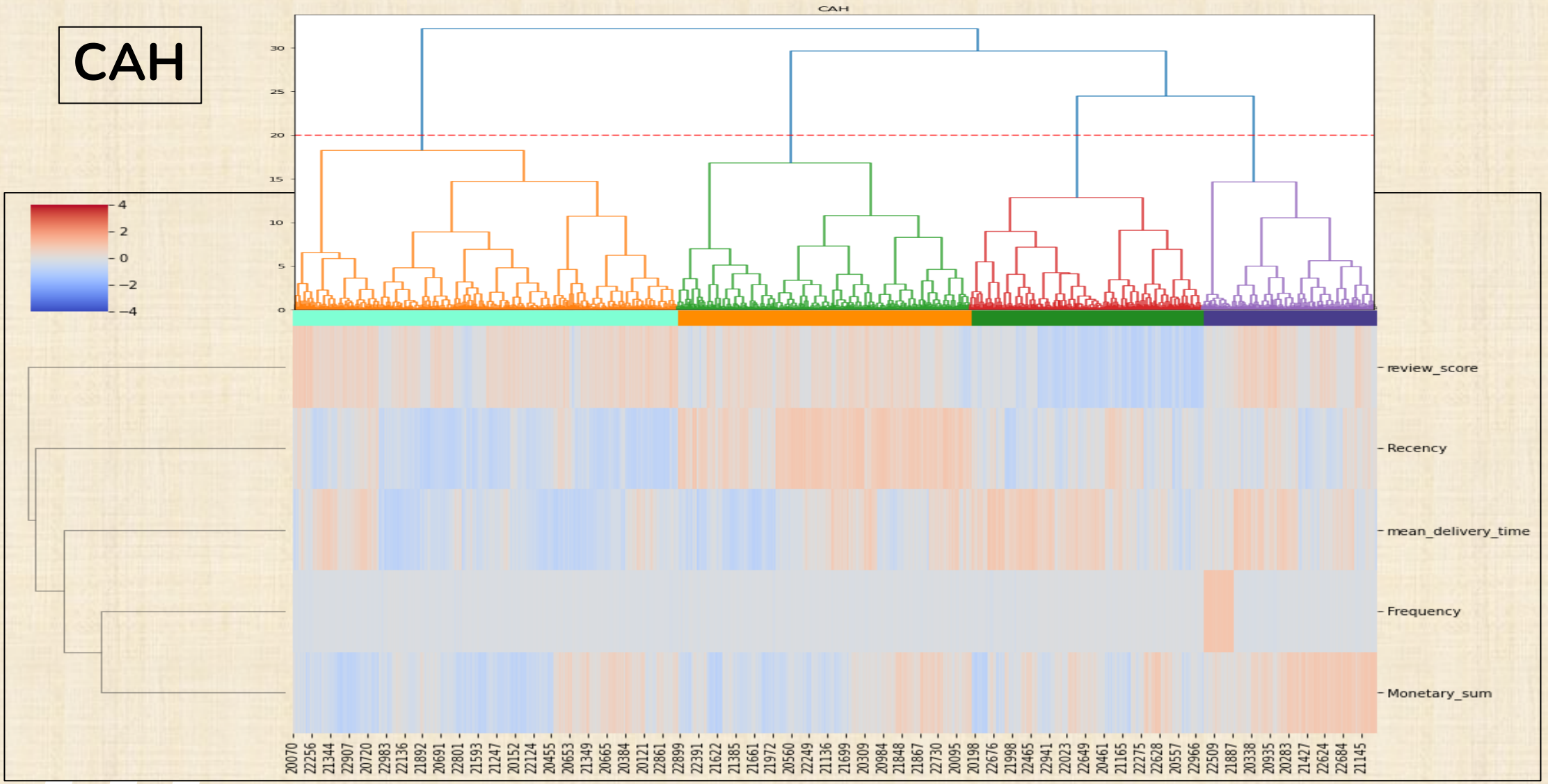


Clusters 4 : des clients avec très faible délais de livraison, au niveau de leurs avis sont super contents « très bonne note d'avis », avec un nombre de commandes moyen. Des commandes passées récemment, avec des montants dépensés faibles.

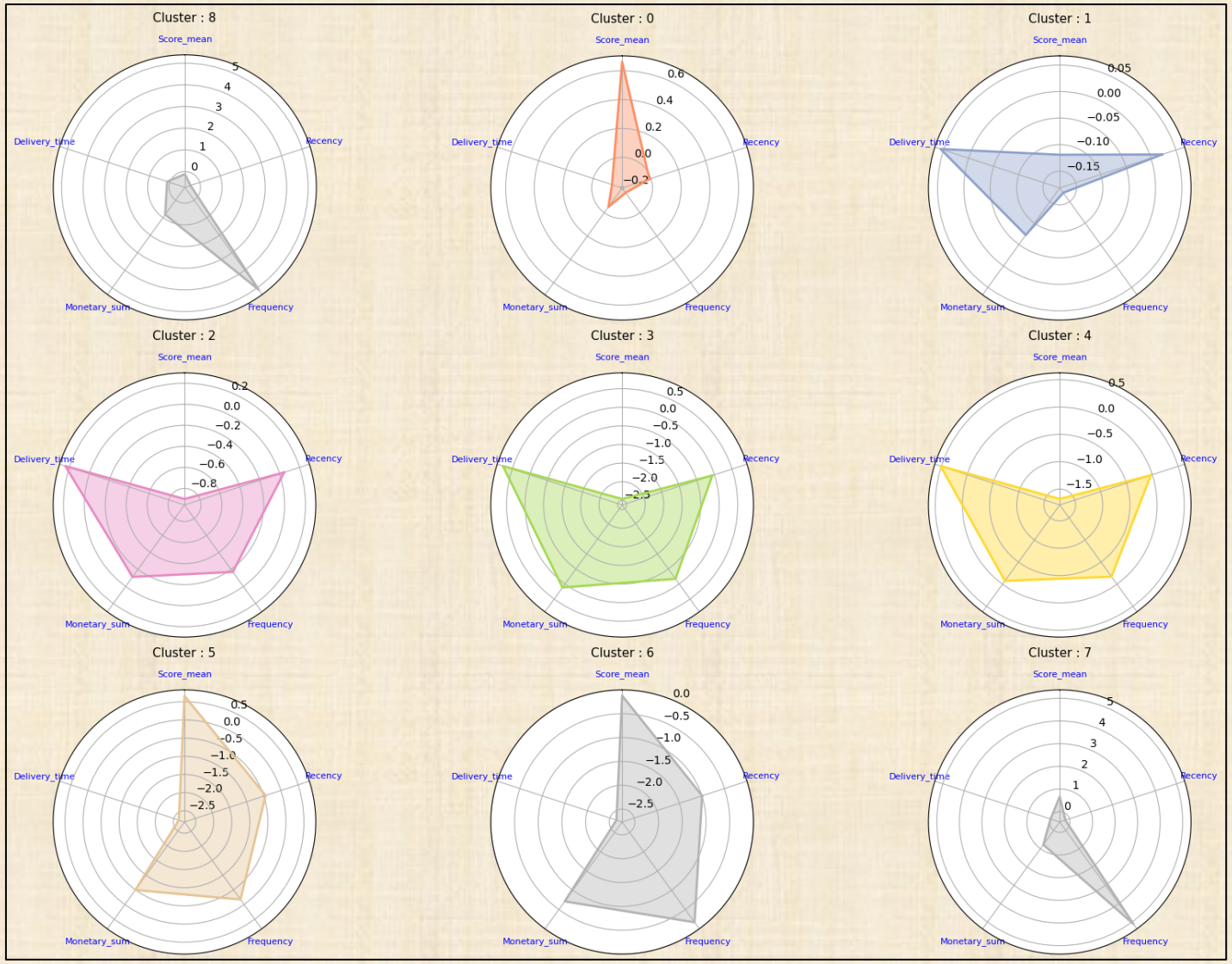
Modélisation 2

olist

CAH

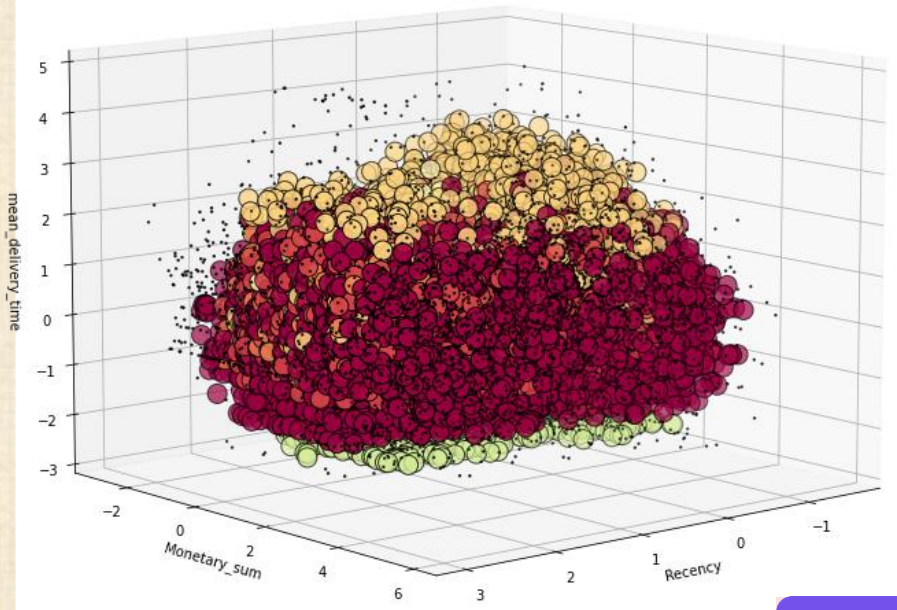


Modélisation 2




```
Counter({0: 51712, 1: 17396, 3: 8359, 2: 7083, -1: 3135, 4: 2314, 7: 1050, 5: 1037, 8: 505, 6: 142})
        Recency  Frequency  Monetary_sum  Delivery_time  Score_mean
72     0.892554  -0.050982      0.094336       0.436554   -0.035640
112    0.272951  -0.045684      0.000659       0.944433    0.177363
124   -0.407537  -0.054765      0.864341       0.014121   -0.289187
146   -0.328358  -0.061637      0.863313       0.192708   -0.325477
216    0.106934  -0.064062      0.133436      -0.753330   -0.631783
...         ...        ...           ...            ...         ...
92625 -0.330118  -0.043908     -0.152222      -0.680985   -0.634178
92636  0.906163  -0.052218     -0.010484      -0.367332    0.202728
92689  0.325310   0.916992     -0.073874       0.147338   -0.161658
92706  0.513353  -0.037892      0.488017       0.444238   -0.547289
92728  0.327238  -0.040648      0.857616       0.361737    0.157809

[3135 rows x 5 columns]
Nombre de Clusters = 9
```

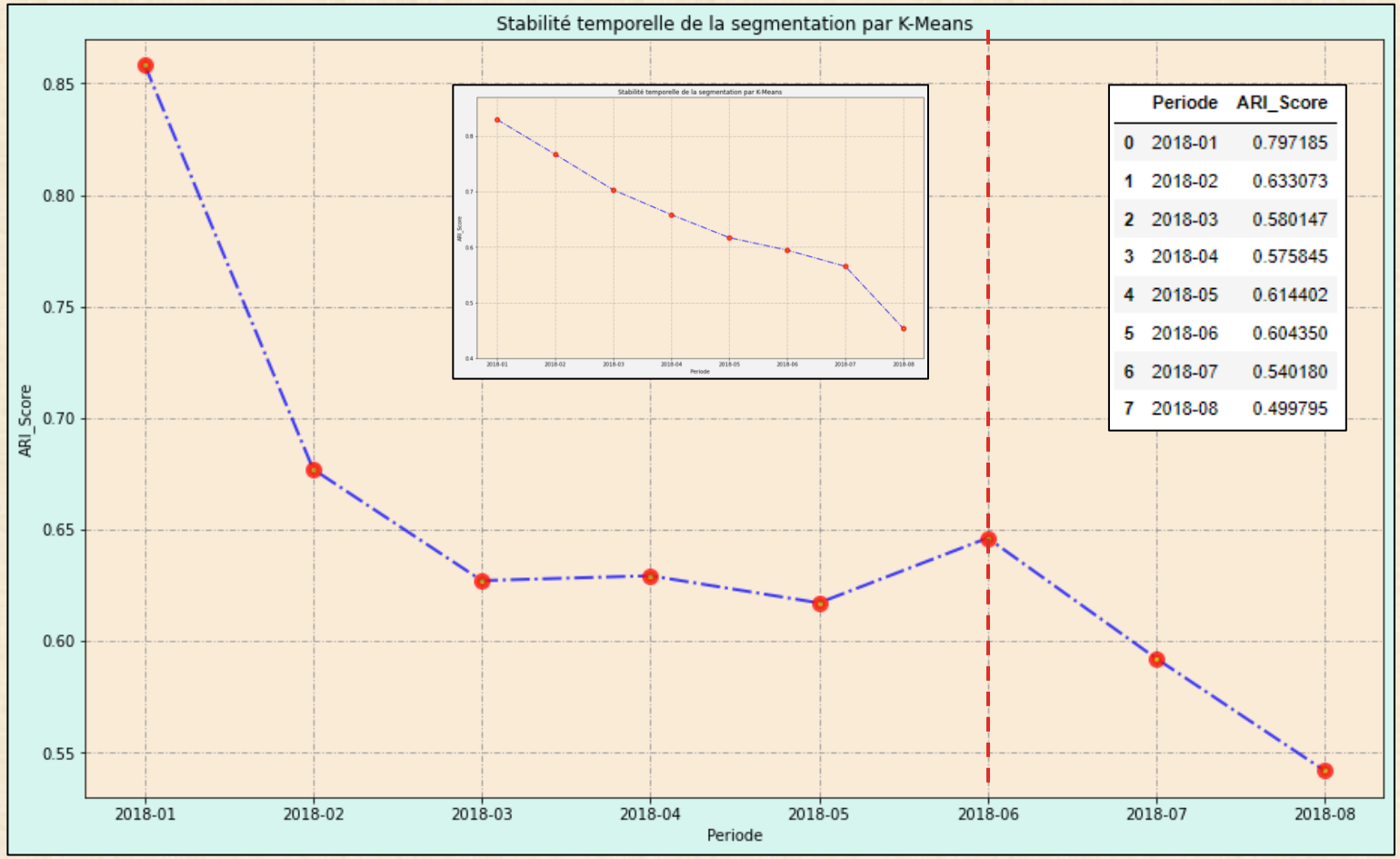


Modèles retenu et Stabilité

Modèles retenu et Stabilité: Kmeans, nombre de clusters = 5

Sur ce plot des scores ARI obtenus sur les itérations par période de 1 mois, on remarque une forte chute après 6 mois sur les clients initiaux.

Il faudrait donc prévoir la maintenance du programme de segmentation tous les 6 mois dans un premier temps puis retester cette stabilité temporelle au fil du temps afin de l'affiner. Il sera donc nécessaire de redéfinir les segments clients à chaque maintenance.



	Periode	ARI_Score
0	2018-01	0.797185
1	2018-02	0.633073
2	2018-03	0.580147
3	2018-04	0.575845
4	2018-05	0.614402
5	2018-06	0.604350
6	2018-07	0.540180
7	2018-08	0.499795

Conclusion

- Test de différents algorithmes dont K-Means, Hierarchiqual agglomerative clustering et DBSCAN, le nombre de cluster tournent autour de 4 et 9 clusters pour les deux parties de modélisation 1 et 2.

- Passage au Log pour certaines variables

- Modélisation 1: écartement de la segmentation RFM car la présentation se fait avec un nombre limité de features ce n'est pas possible d'ajouter d'autres critères, ainsi que le choix arbitraires des scores, choix de l'algorithme non supervisé K-Means pour enréchir la segmentation et éviter les choix arbitraires .

- Maintenance et choix d'algorithme:
  - K-Means (initialisation facile , Nbs de features non limité, pas de d'attribution arbitraire de score)
  - Fréquence de MàJ chaque 6 mois selon le score ARI.