Segmentez des clients d'un site e-commerce

Mitra Dadgar

01

Introduction

Définition du problème, présentation des données

02

Analyse descriptive

Analyses des comportements clients, RFM

03

Les segmentations

K Means, DBScan, Agglomerative Hierarchical Clustering

04

Conclusion

Stabilité,maintenance

Introduction

Une entreprise brésilienne qui propose une solution de vente sur les marketplaces en ligne

Objectifs :

- ➔ Comprendre les différents types d'utilisateurs (une segmentation des clients RFM)
- ➔ Fournir aux équipes e-commerce une segmentation des clients pour les campagnes de communication
- ➔ Proposer un contrat de maintenance basé sur une analyse de la stabilité des segments au cours du temps.

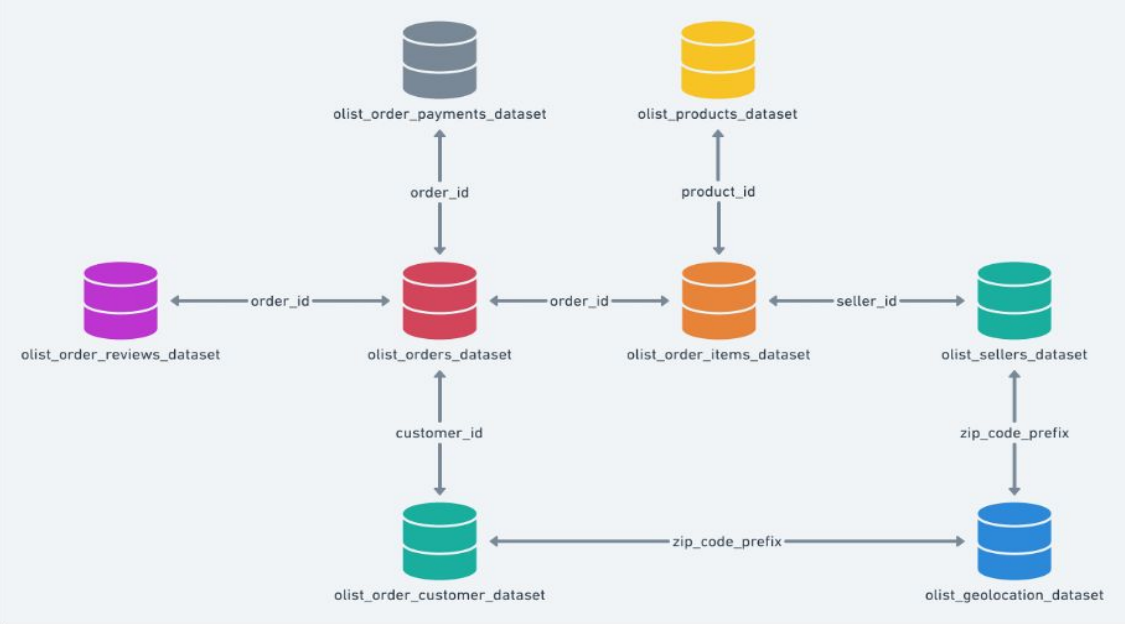
Problématiques:

- ➔ Exploration des données
- ➔ Choisir des features adaptées
- ➔ Trouver une classification non supervisée qui être explicable et réutilisable pour des campagnes de communication

Présentation des données

Données réparties en 9 tables:

(clients / géolocalisation / commandes / paiements / produits / vendeurs / traduction des catégories de produits)



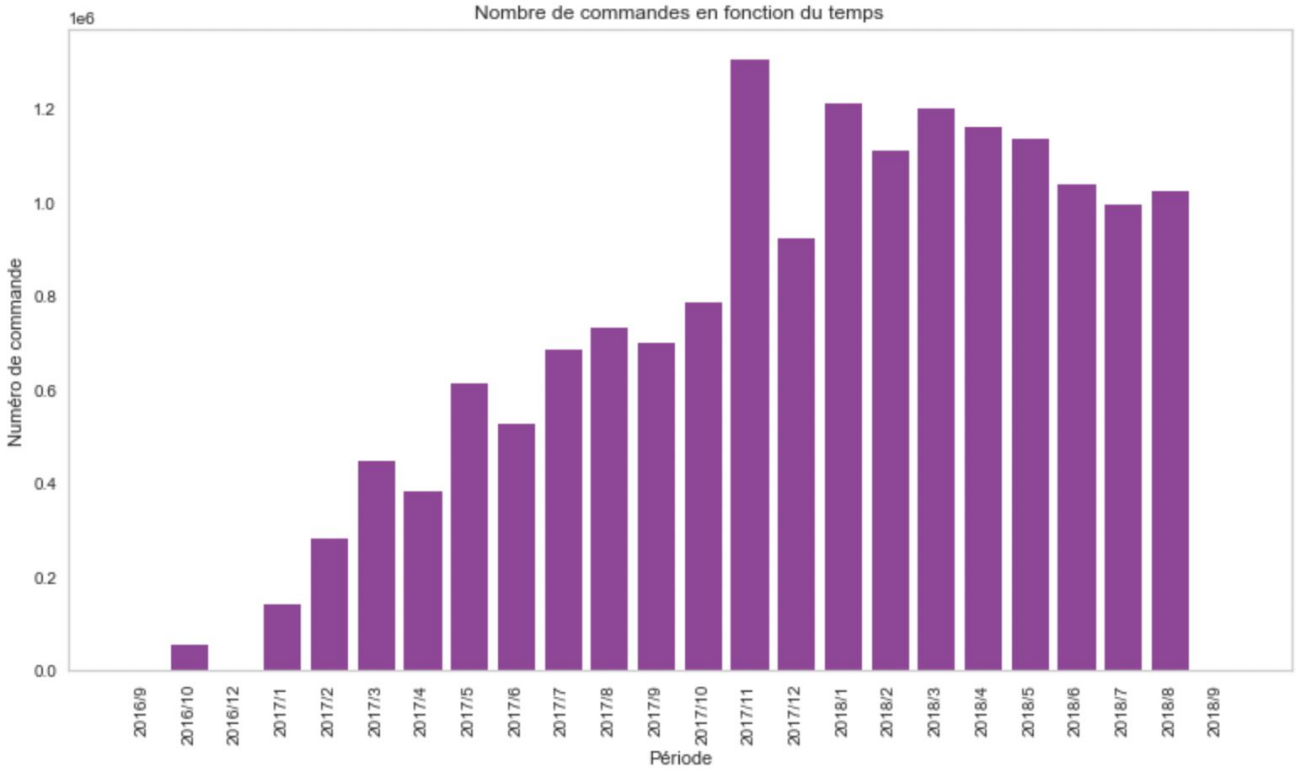
Analyse descriptive

●Analyse de clients
- Nombre d'id clients: 96516
- Nombre d'id unique: 93396

●Analyse de la fréquence
- Un achat chez 90% des clients

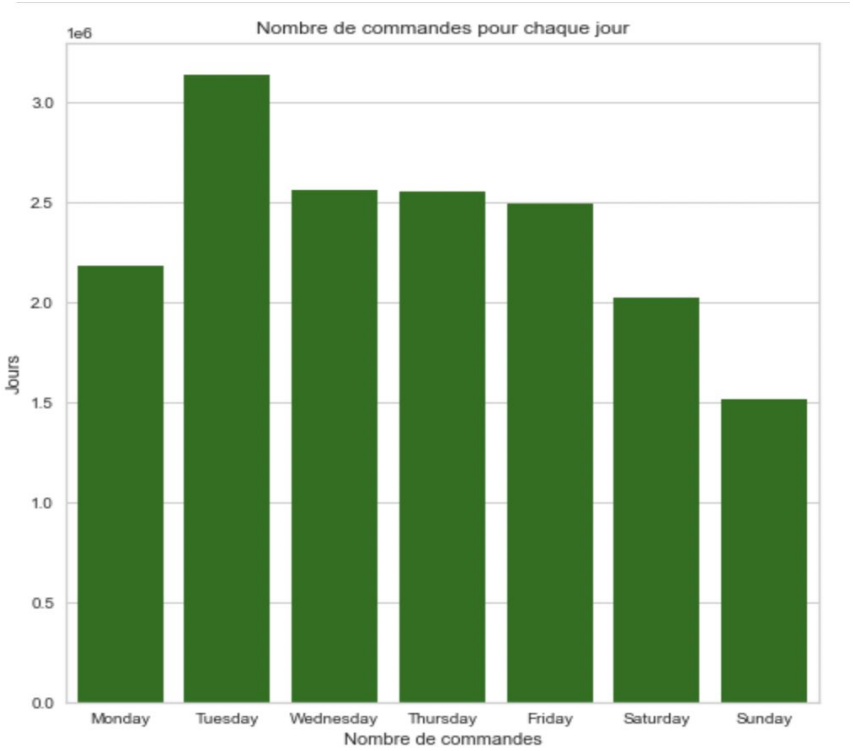
●Analyse de la paiment
- La plupart des commandes sont livrées
- Les clients ont payé en une seule fois et le nombre de paiment est plus d'une fois
- Les commandes coûtent moins de 2000

Barplot répartition de commandes en fonction des mois

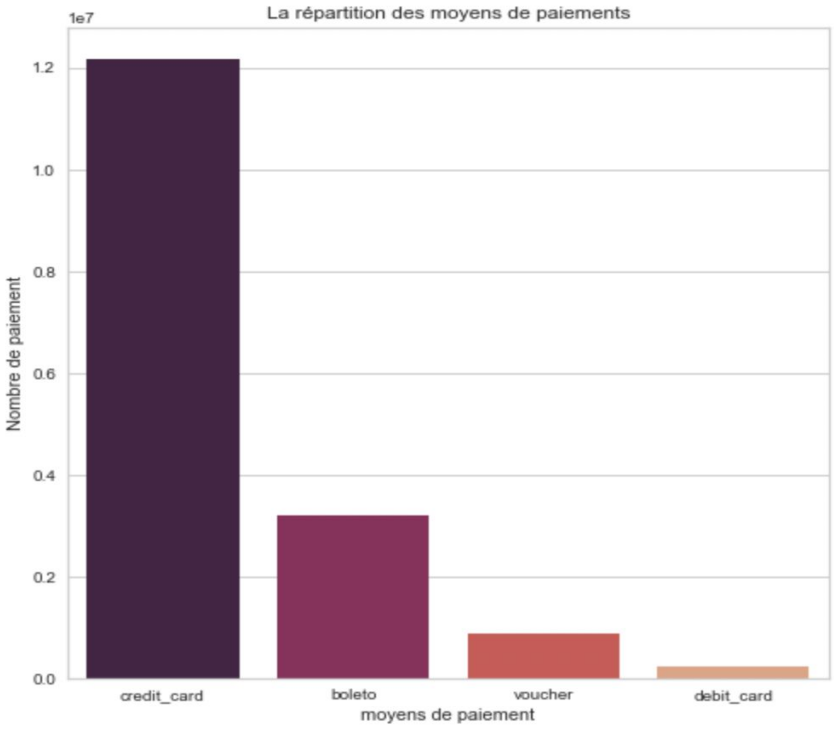


- Le nombre de ventes a considérablement augmenté en 2017
- En novembre 2017 les ventes ont augmenté mais en décembre 2017 et juin 2018, les ventes ont chuté

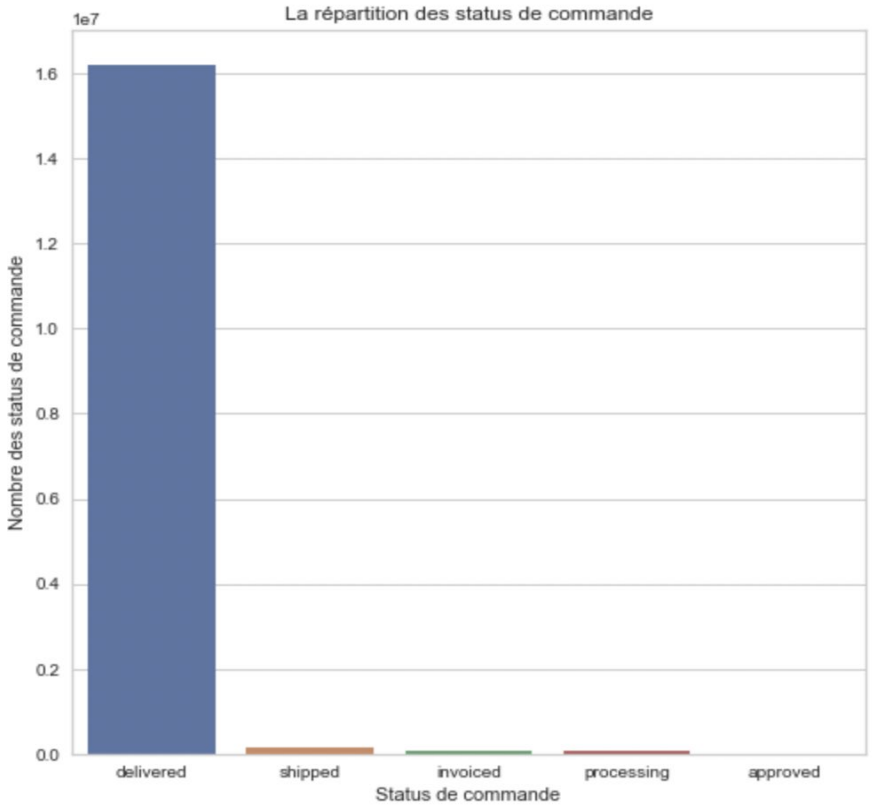
Barplot répartition de commandes en fonction des jours



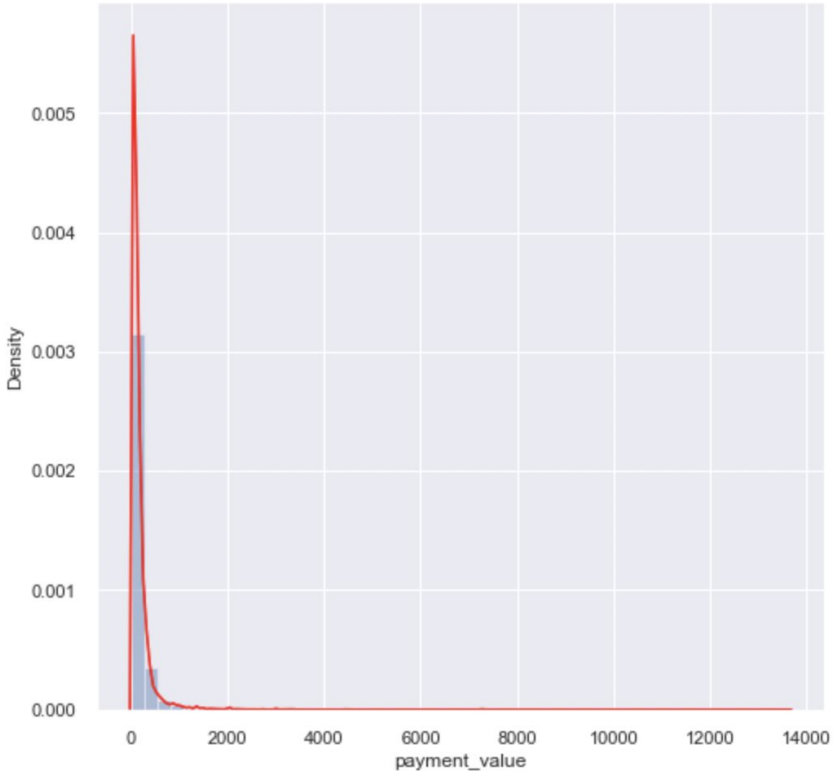
Barplot répartition des moyens de paiements



Barplot répartition des statuts de commande



Histogramme montant des commandes



Analyse RFM

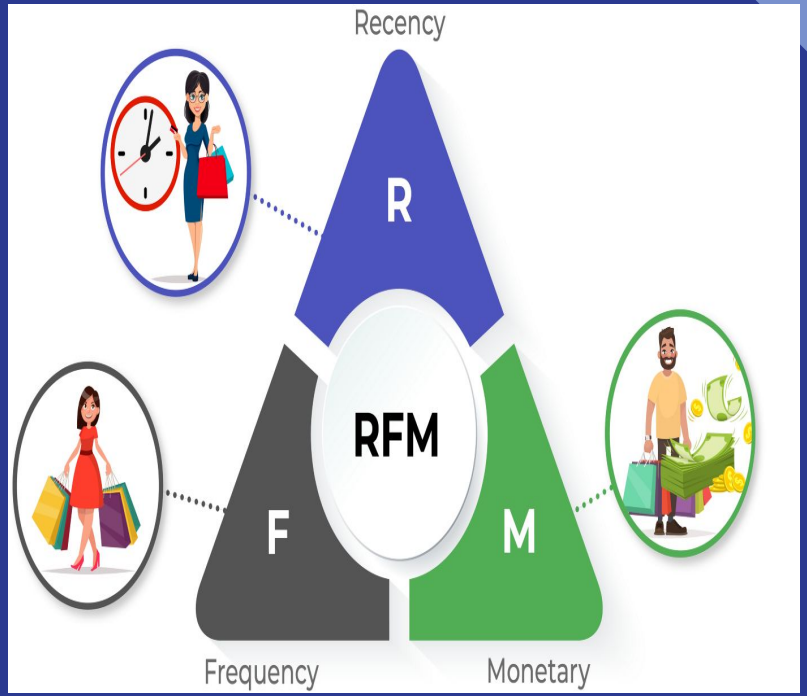
L'utilisation analyse RFM pour segmentation du comportement des clients basée sur les données.

Pour le RFM, nous aurons besoin de 3 indicateurs :

- la récence (Recency),
- la fréquence (Frequency),
- l'argent moyen dépensé par client (Monetary) => 'payment_value'.
- Pour la récence et le délai de livraison, nous utiliserons ces 2 fonctionnalités => 'order_purchase_horodatage' 'order_delivered_customer_date'

Etapes:

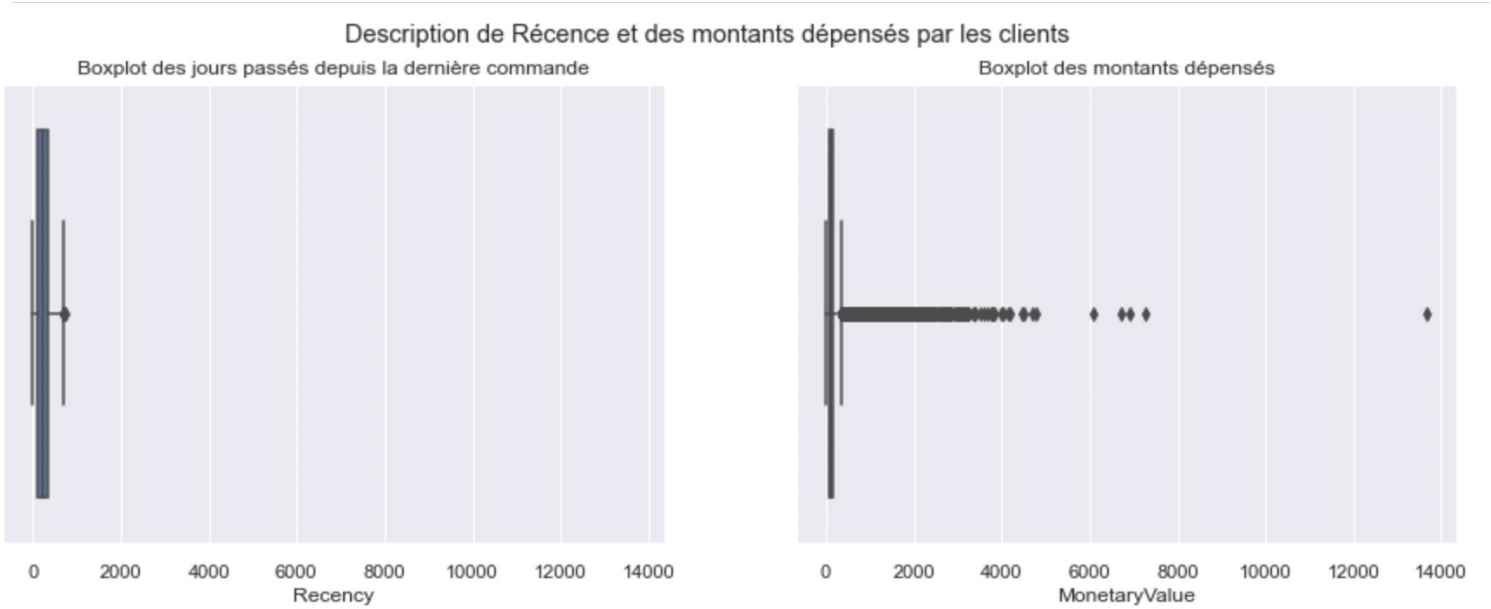
1. Transformation des horodateurs en datetime
2. Calcul des scores à l'aide de quartiles
3. Création RFM Score avec la concaténation des quartiles/bins R, F M
4. Analyse de la segmentation RFM



RFM

customer_unique_id	Recency	Frequency	MonetaryValue	r_quartile	f_bins	m_quartile	RFM_Score
4d99682572b7b5932340a0bce676c18c	87	4872	244.020000	1	1	1	111
96e91c0dba30f7ff60c9acd47677c248	47	4992	1351.440000	1	1	1	111
397b44d5bb99eabf54ea9c2b41ebb905	78	6783	490.867143	1	1	1	111
86bfc49565a9ca52fcbf861fcc1e67a4	83	6979	152.492857	1	1	2	112
5419a7c9b86a43d8140e2939cd2c2f7e	114	6072	55.575000	1	1	4	114
...	...	...	...	...	...	...	...
876a8881e182fb3a64af97b737cfe889	502	227	39.420000	4	2	4	424
876531f7c6bb88815b1ae79c3ad4719a	427	102	33.770000	4	2	4	424
875e9bc9dba22c41f1931a2d55201ec1	519	55	57.720000	4	2	4	424
ea715f021baf4f64753e33d7b288b72b	467	80	55.000000	4	2	4	424
2f6bfa6509c44c47a19d724930a66a29	421	73	47.680000	4	2	4	424

Analyse de la segmentation RFM

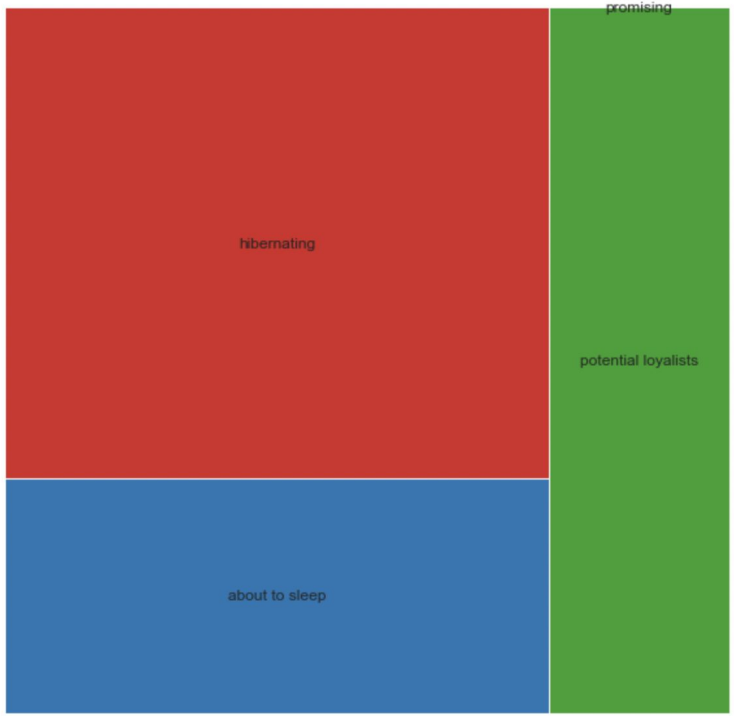


1: La plupart des clients réalisent un seul achat.
2: Les sommes dépensées atteignent un maximum de 14.000 Réaux.
3: Il est difficile de savoir s'ils reviennent régulièrement acheter donc la récence est très variable. On voit que la fréquence est inutile, il faut donc réfléchir à l'ajout de 2 nouvelles variables.

Nommer les segments et attribuer les individus

Méthode 1:
Les scores RFM nous donnent 533 = 125 segments. Ce qui n'est pas facile à travailler. Je vais travailler avec 11 segments basés sur les scores R et F. Voici la description des segments

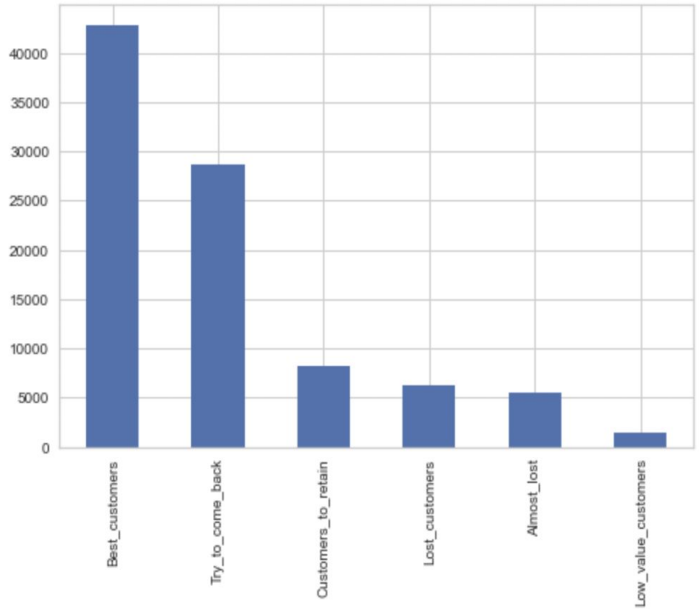
Segment	Description
Champions	Acheté récemment, achetez souvent et dépensez le plus
Loyal Customers	Achetez régulièrement. Réactif aux promotions
Potential Loyalist	Clients récents avec fréquence moyenn
Recent Customers	Acheté le plus récemment, mais pas souvent
Promising	Acheteurs récents, mais qui n'ont pas dépensé beaucoup
Customers Needing Attention	Récence, fréquence et valeurs monétaires supérieures à la moyenne. Peut-être pas acheté très récemment cependant
About To Sleep	Récence et fréquence inférieures à la moyenne. Les perdront s'ils ne sont pas réactivés
At Risk	achete souvent mais il y a longtemps. Faut les ramener !
Can't Lose Them	Utilisé pour acheter fréquemment mais n'est pas revenu depuis longtemps
Hibernating	Le dernier achat remonte à longtemps et le nombre de commandes était faible. Peut être perdu

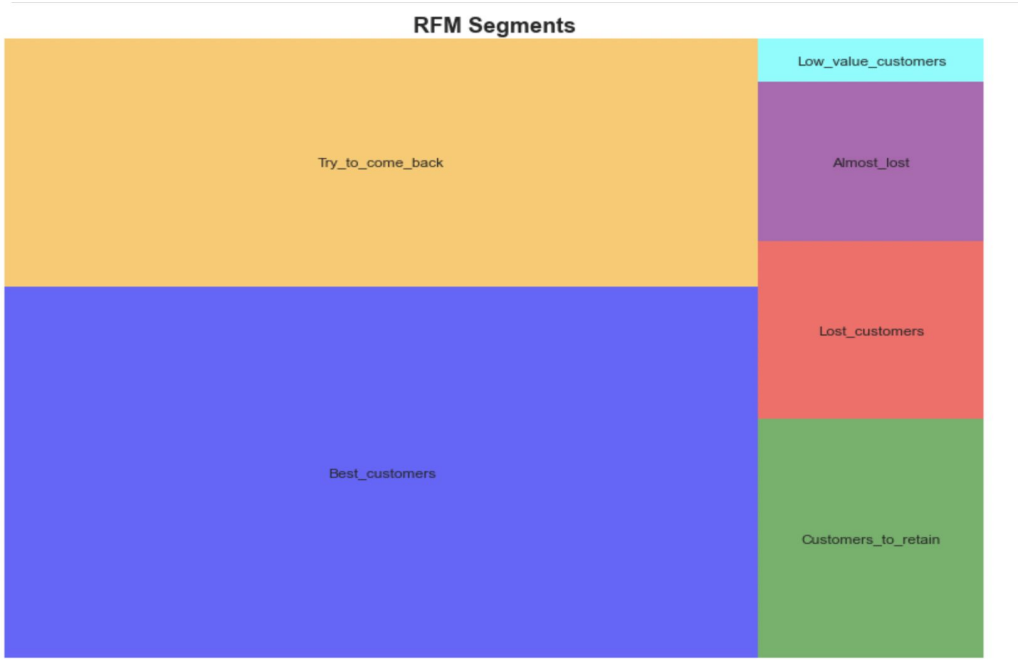


Méthode basée sur du scoring

- **Méthode 2:** On fait une distribution normale pour ‘payment_value' a Les scores seront de 1, 2 ou 3 pour chaque élément de chaque ligne. 1 sera un mauvais score. 3 sera le meilleur score possible.
- 'Frequency' : En raison de la distribution spéciale, la répartition est arbitraire . Fréquence = 1 = mauvaise note (1) . Fréquence 2 = meilleur score (3)
- 'Recency' : En raison de la distribution normale, nous pouvons diviser la distribution en 3 scores

```
data_rfm['RFM_Score2'] = data_rfm['recency_score_bis'] * 100 \
                       + data_rfm['frequency_score_bis'] * 10 \
                       + data_rfm['payment_value_score_bis']
```

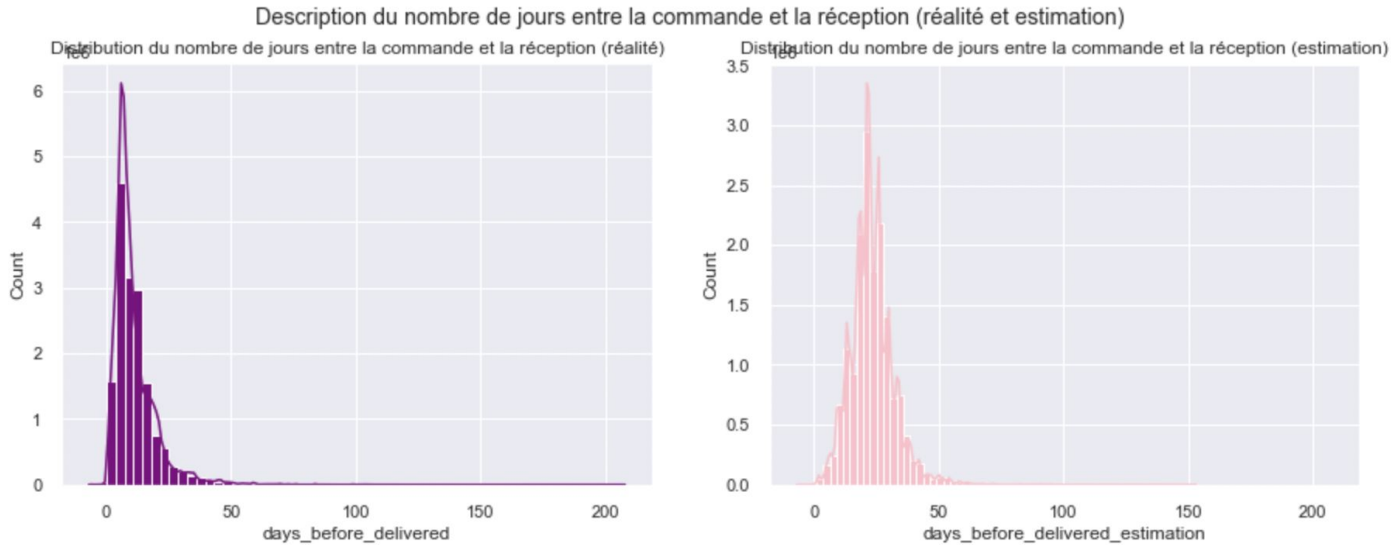




Amélioration des variables

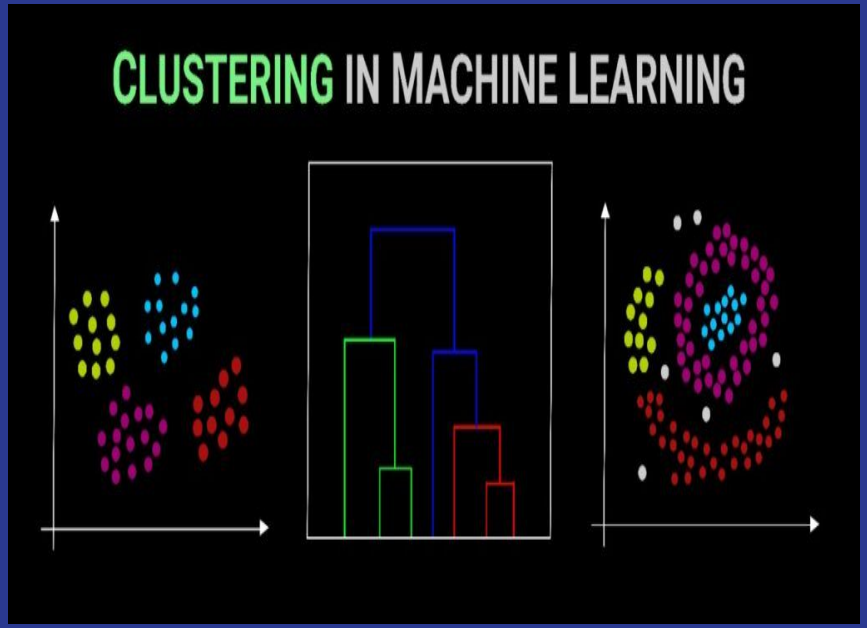
On va créer de nouvelles variables que l'on va ajouter pour obtenir une segmentation plus pertinente.

1. le nombre de jours entre la commande (order approved at) et la réception de cette dernière
2. le nombre de jours entre la date de commande livrée et la date d'estimation



Modélisation :

- K_Means
- DBScan
- Agglomerative Hierarchical Clustering



Modélisation

- **K-Means** : qui se base sur des calculs de distance entre les points de notre jeu de données et un point nommé centroïde

  ➔ Rechercher le K optimal avec le silhouette score

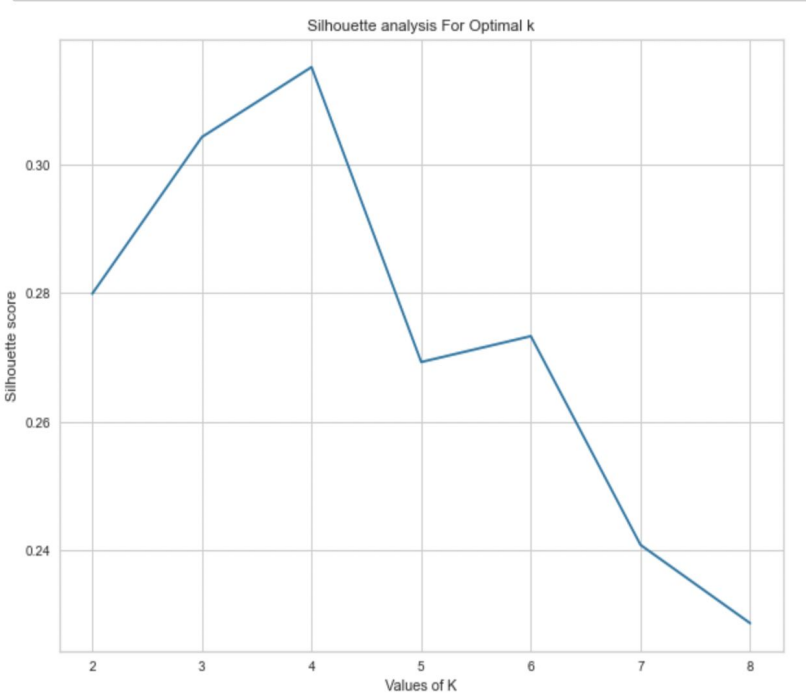
- **DBScan** : considère les clusters comme des zones à haute densité séparées par des zones à faible densité

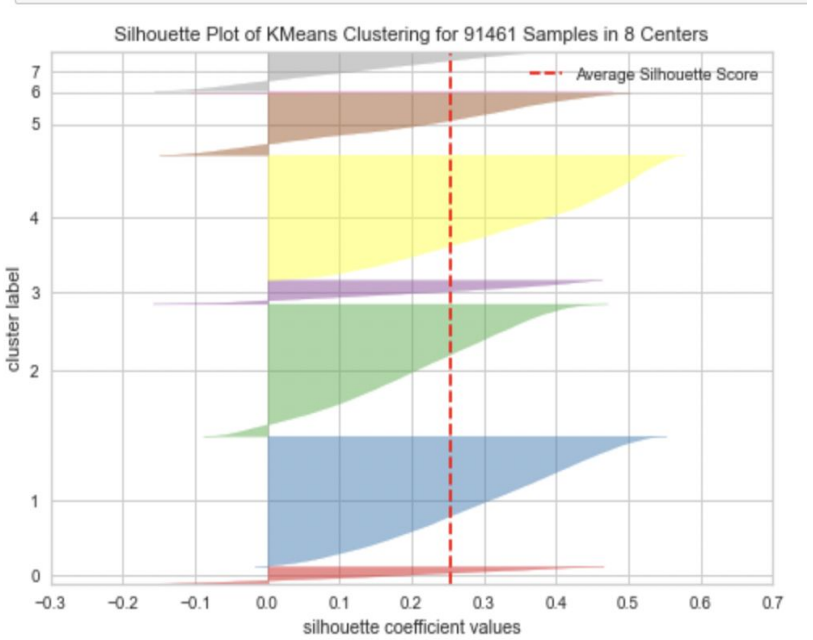
  ➔ Rechercher deux paramètres à l'algorithme, min_samples et eps

- **CAH** : Cette hiérarchie de clusters est représentée sous forme d'arbre (ou dendrogramme). La racine de l'arbre est la grappe unique qui rassemble tous les échantillons, les feuilles étant les grappes avec un seul échantillon

  ➔ Le critère linkage : (single, average, ward)

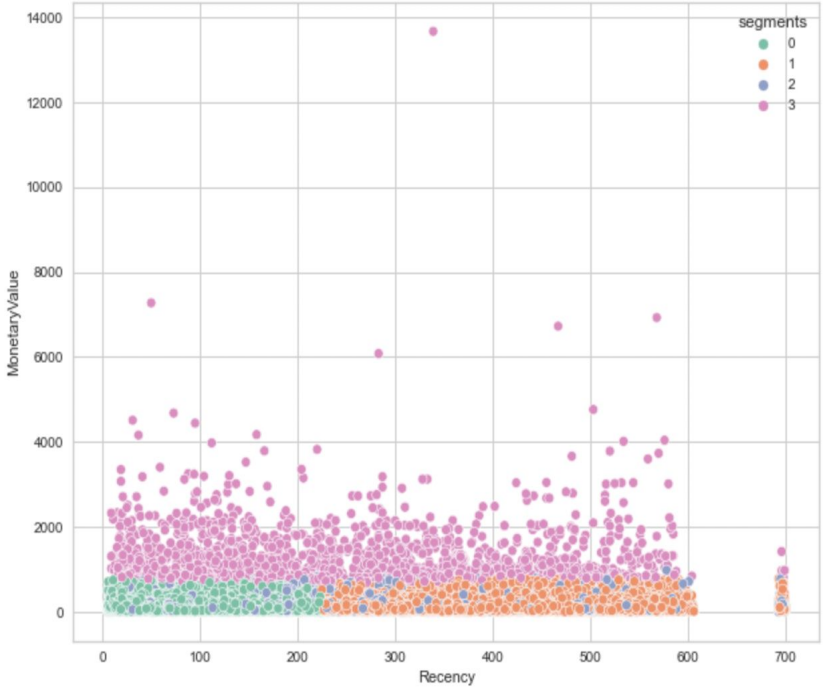
Modélisation : K-Means



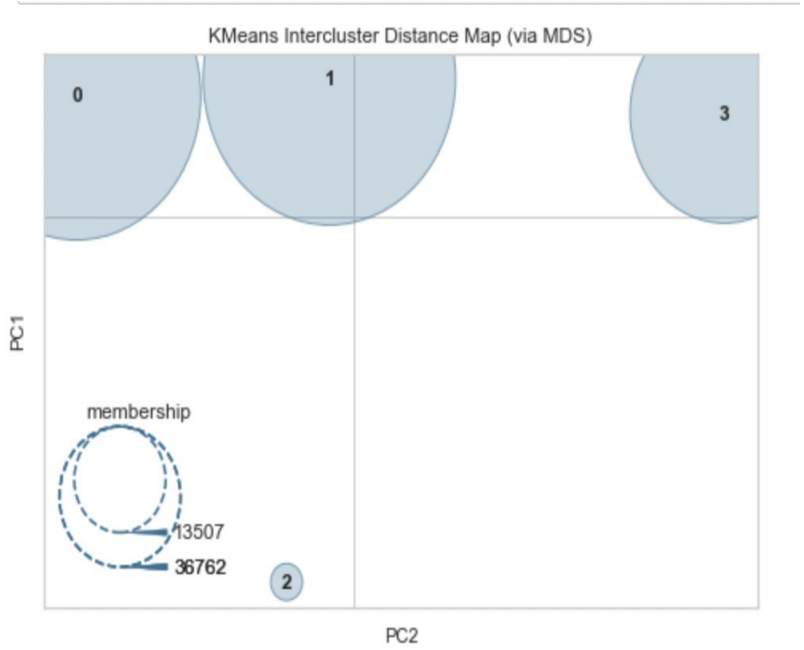


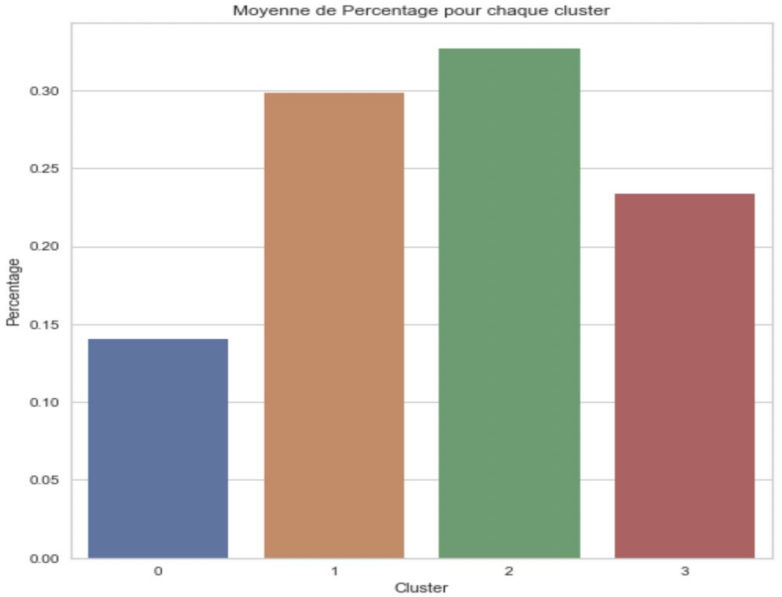
K-Means


Représentation des clusters du K-Means
0
1
2
3
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
0
5
10
15
20

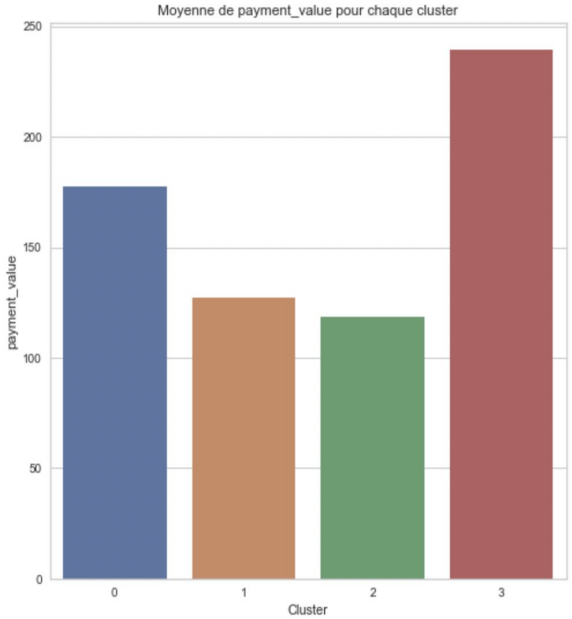

segments
0
1
2
3
MonetaryValue
14000
12000
10000
8000
6000
4000
2000
0
0
100
200
300
400
500
600
700
Recency

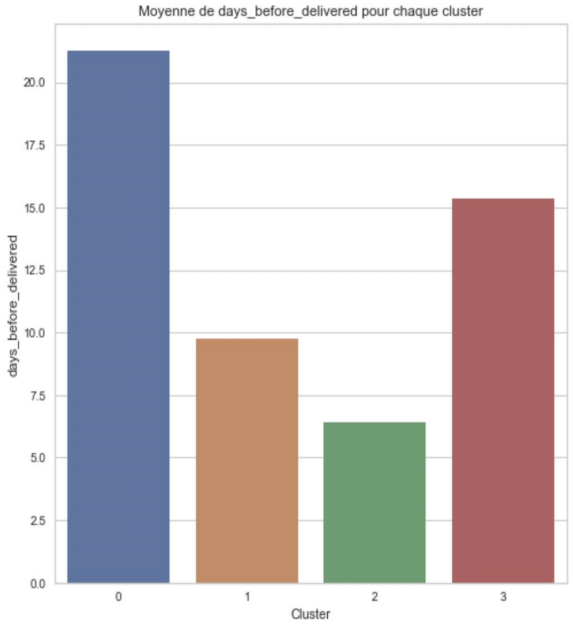
Interpréter les segments en calculant la moyenne des variables par cluster

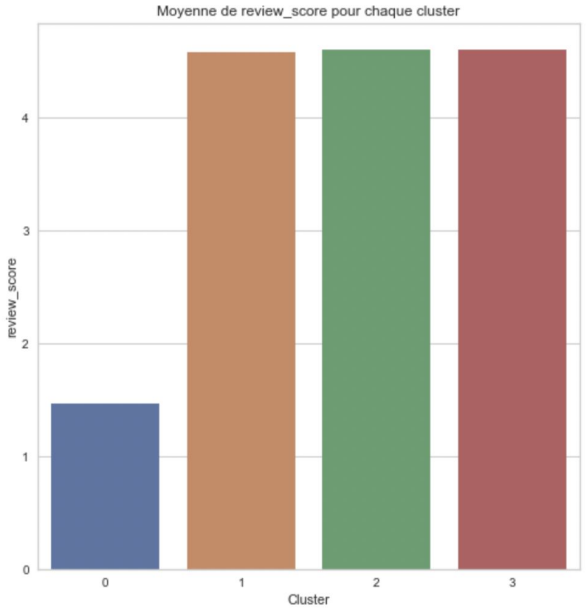




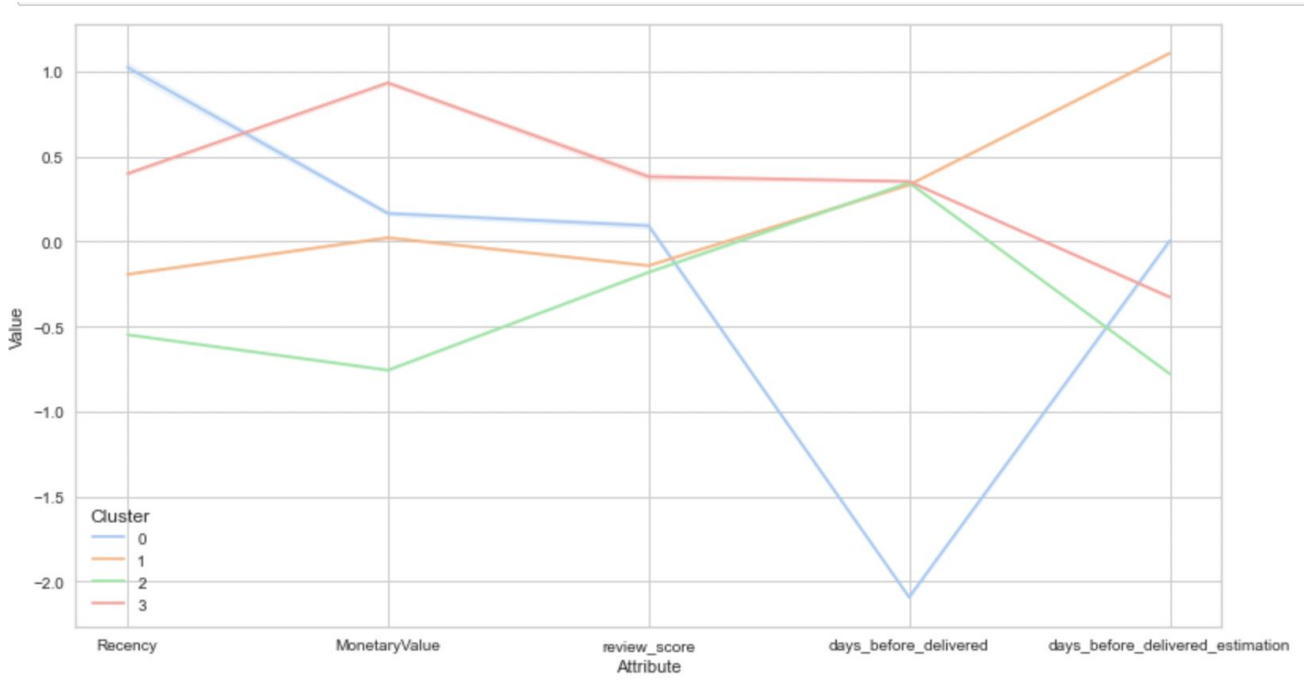
Interpréter les segments en calculant la moyenne des variables par cluster





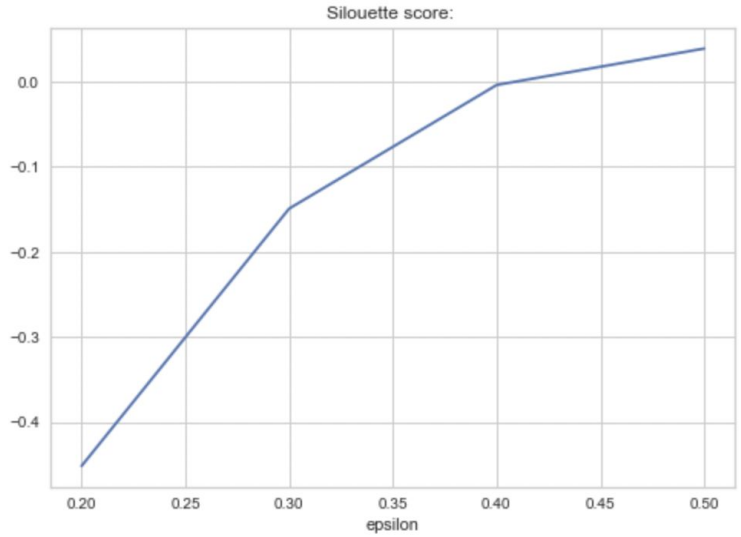


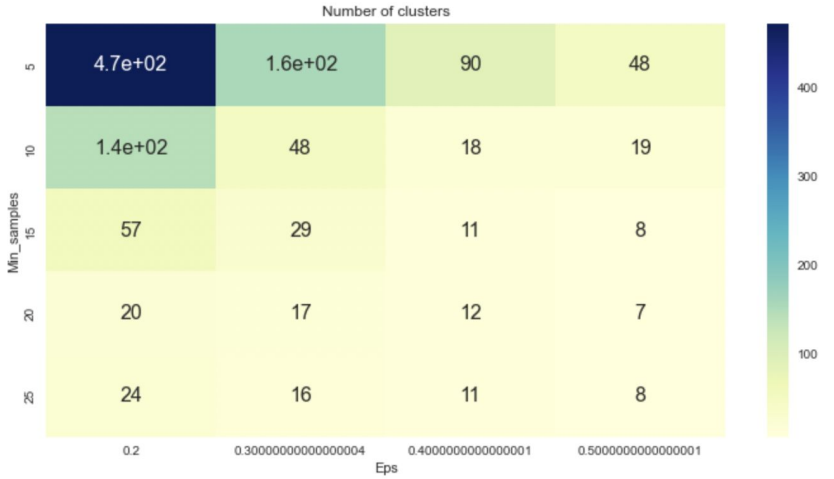
Visualisation parallel plot des valeurs moyennes par cluster

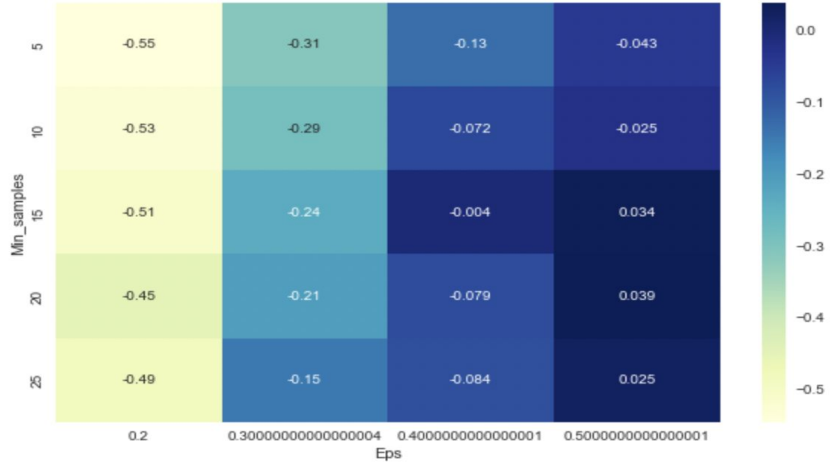


- 0 : les clients satisfait et récent avec un minimum jour de la livraison
- 1 : les clients qui sont pas trop satisfait à cause de retard de livraison
- 2 : les clients pas satisfait et qui ont un petit paniers avec un très long délai de livraison
- 3: les clients gros acheteurs et plus satisfait qui ont acheté les paniers normaux

Modélisation : DBSCAN



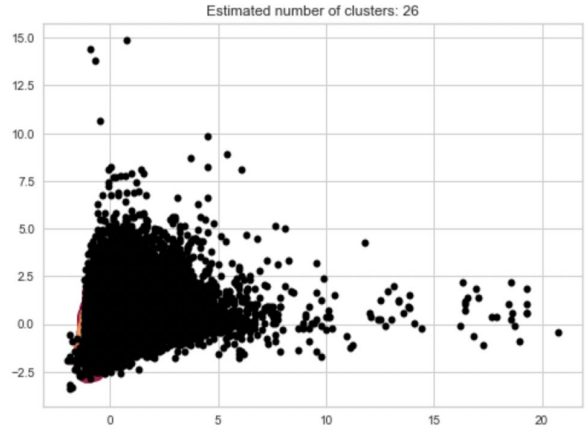




- La carte de chaleur montre que le nombre de clusters varie de 90 à 7. Cependant, la plupart des combinaisons donnent 8 à 11 clusters. Pour décider quelle combinaison choisir, j'utiliserai une métrique score silhouette.

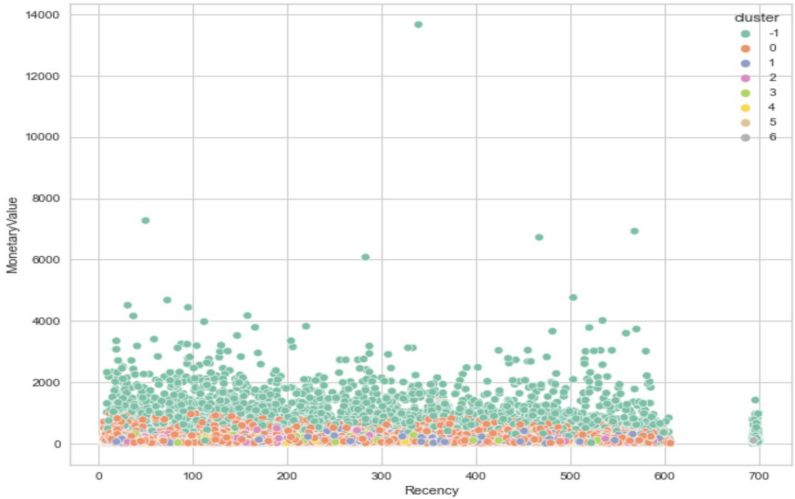
- La deuxième carte de chaleur montre que le maximum global est de 0,039 pour eps=0,5 et min_samples=15.

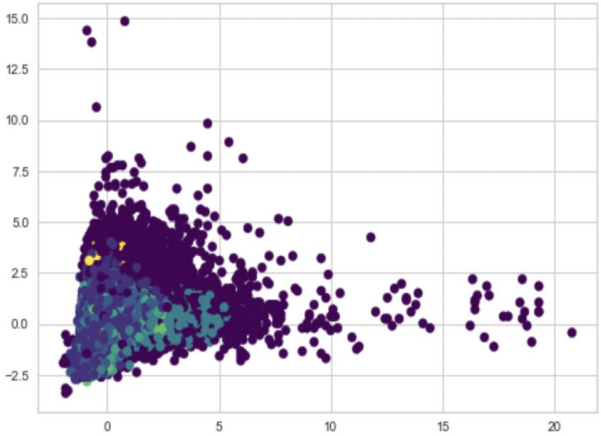
DBSCAN



eps=0.3, min_samples=25

```
Estimated number of clusters: 15
Estimated number of noise points: 26845
Silhouette Coefficient: -0.149
```

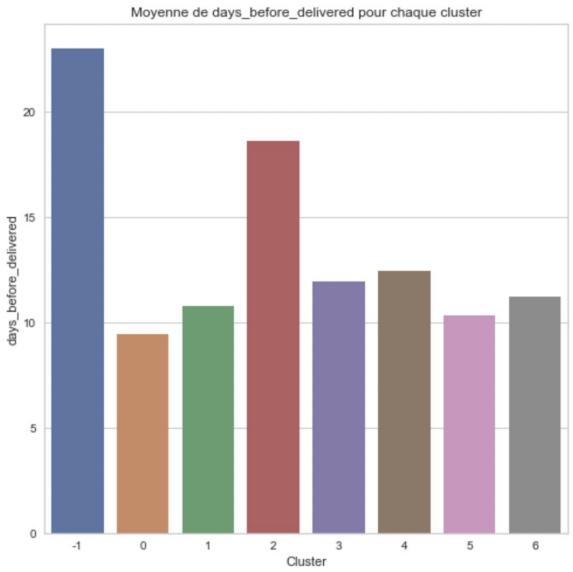


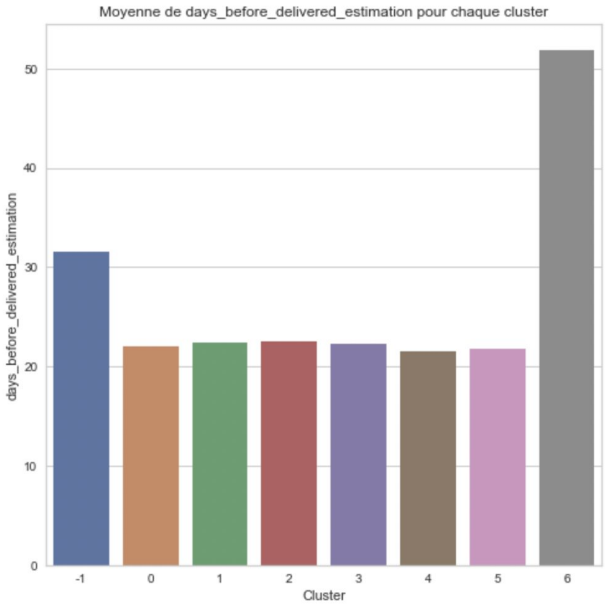


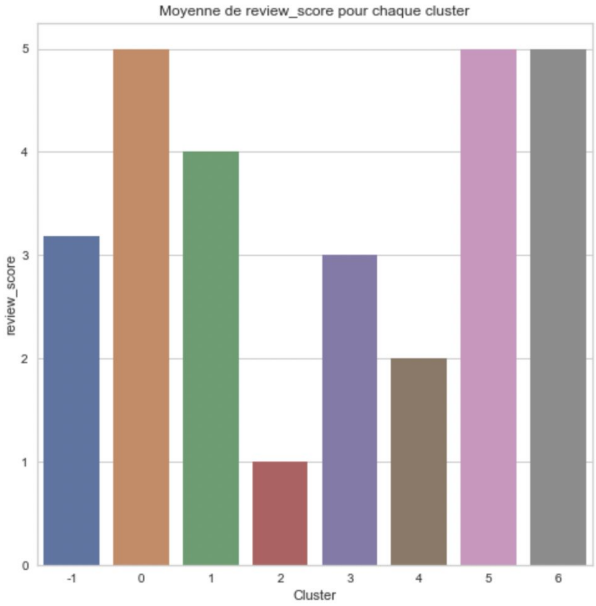
eps=0.5, min_samples=15

```
Estimated number of clusters: 7
Estimated number of noise points: 6300
Silhouette Coefficient: 0.034
```

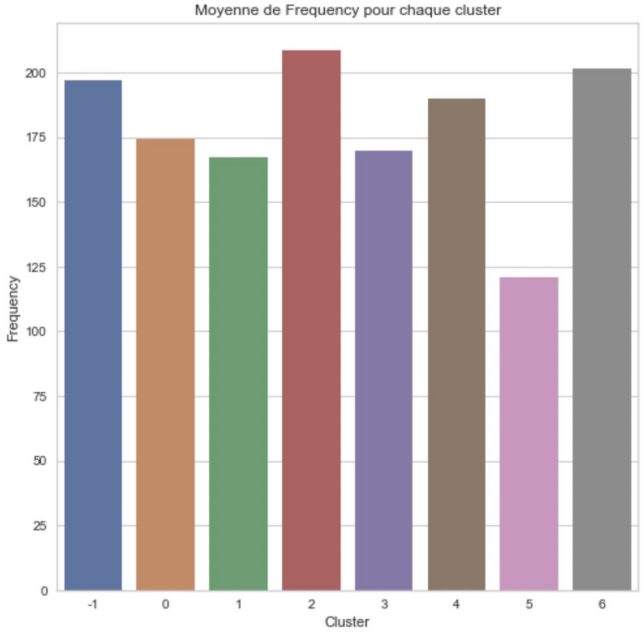
Interpréter les segments en calculant la moyenne des variables par cluster DBSCAN

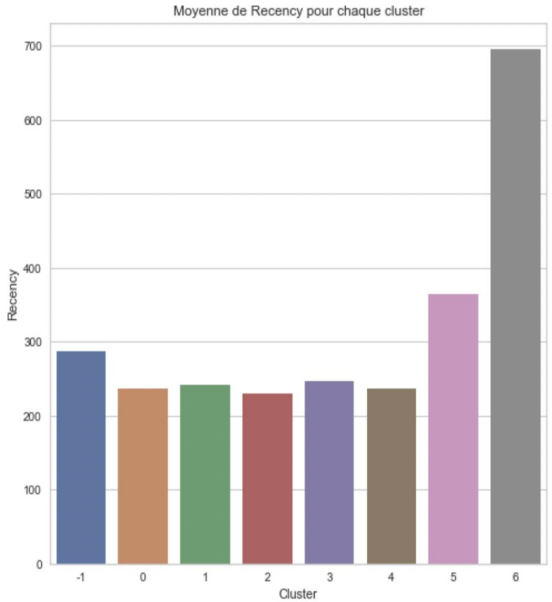


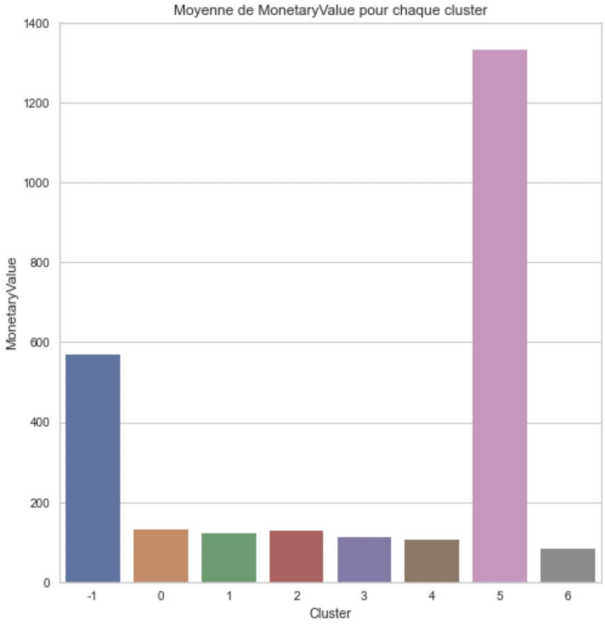




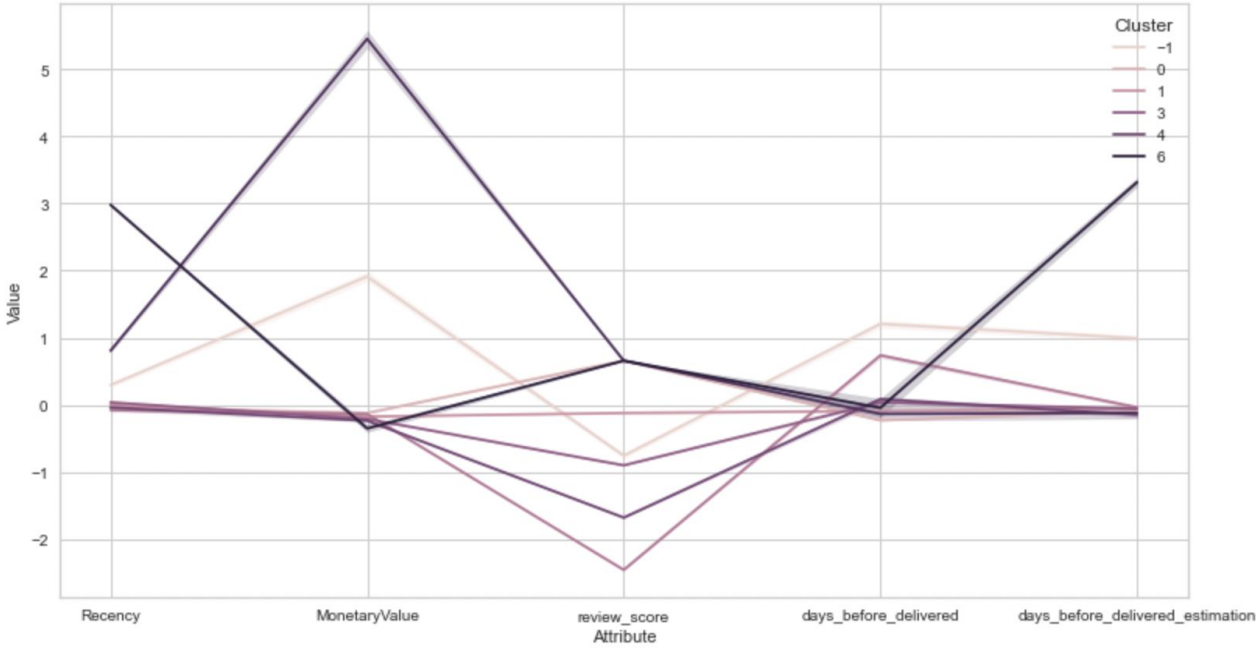
la moyenne des variables par cluster DBSCAN







Visualisation parallel plot des valeurs moyennes par cluster DBSCAN

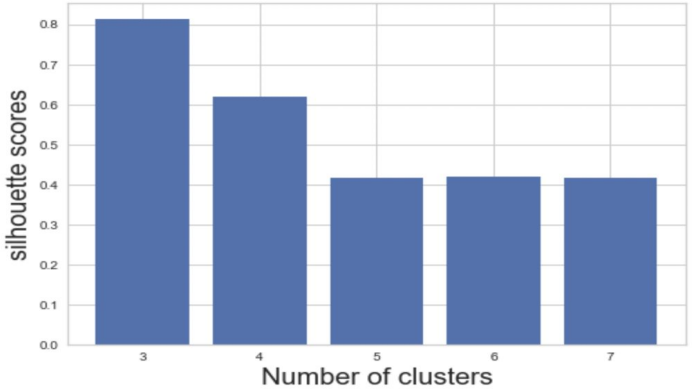


à cause du bruit on ne peut pas bien nommer des segments, mais on voit que :

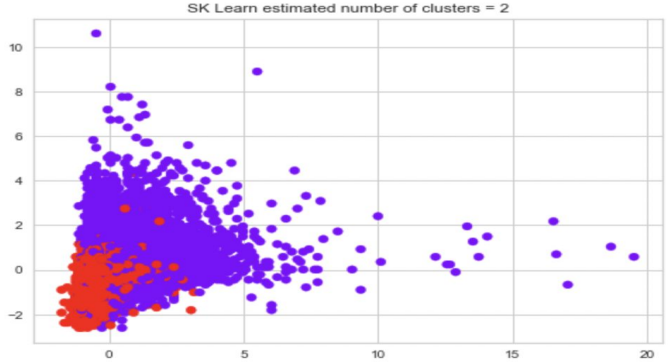
- le segment 6 : acheteurs plus récents avec petit panier et plus satisfait
- le segment 3 : acheteurs insatisfaits avec un très long délai de livraison

Modélisation : CAH

Rechercher k avec le silhouette score

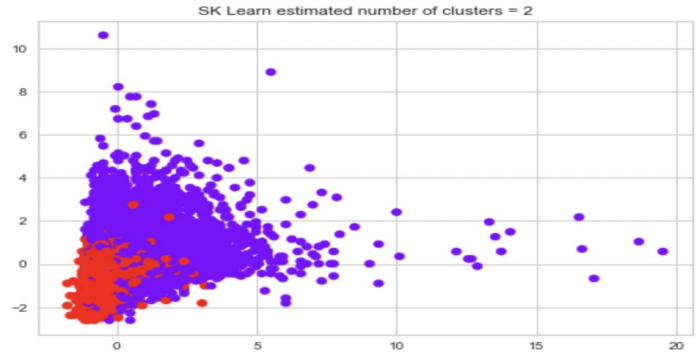



```
linkage is :  average
Number of clusters = 2
Classifying the points into clusters:
[0 1 0 ... 0 1 1]
```

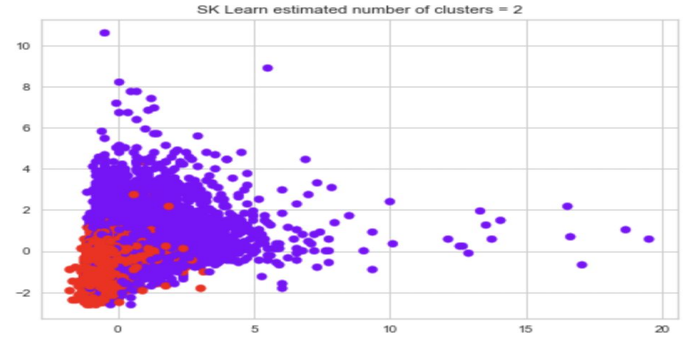


Rechercher le distance threshold avec le silhouette score

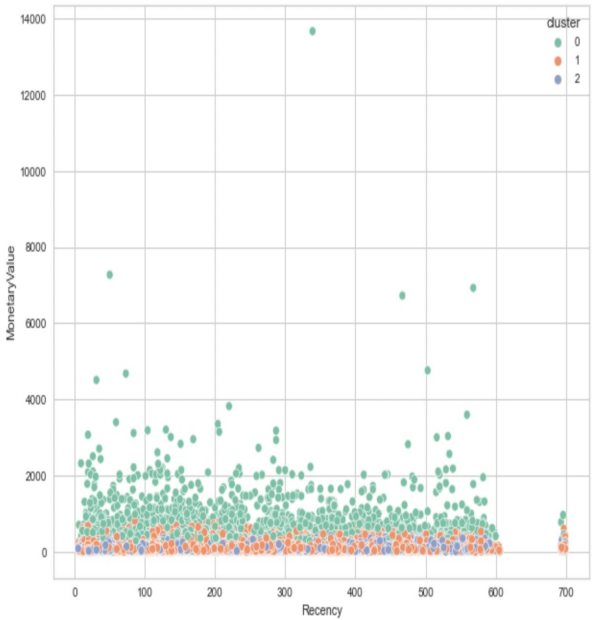
```
linkage is :  single
Number of clusters = 2
Classifying the points into clusters:
[0 1 0 ... 0 1 1]
```

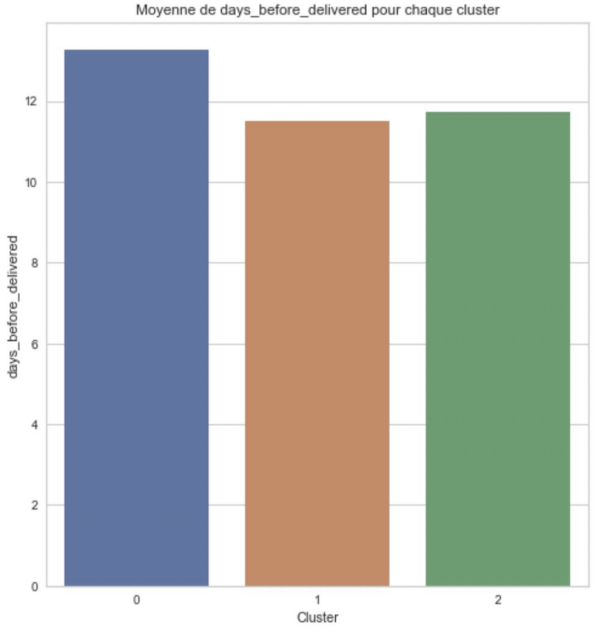


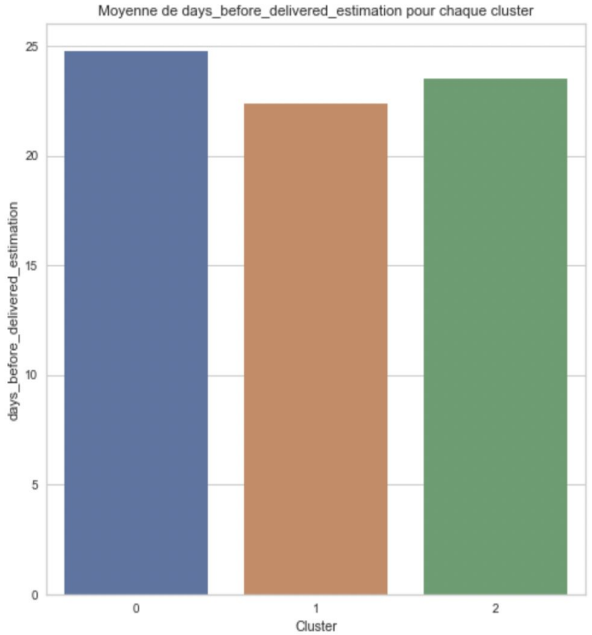

```
linkage is :  ward
Number of clusters = 2
Classifying the points into clusters:
[0 1 0 ... 0 1 1]
```



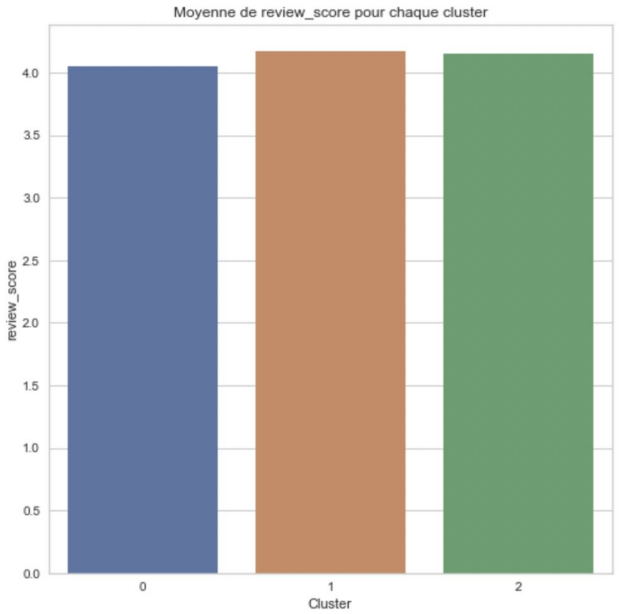
Interpréter les segments en calculant la moyenne des variables par cluster CAH

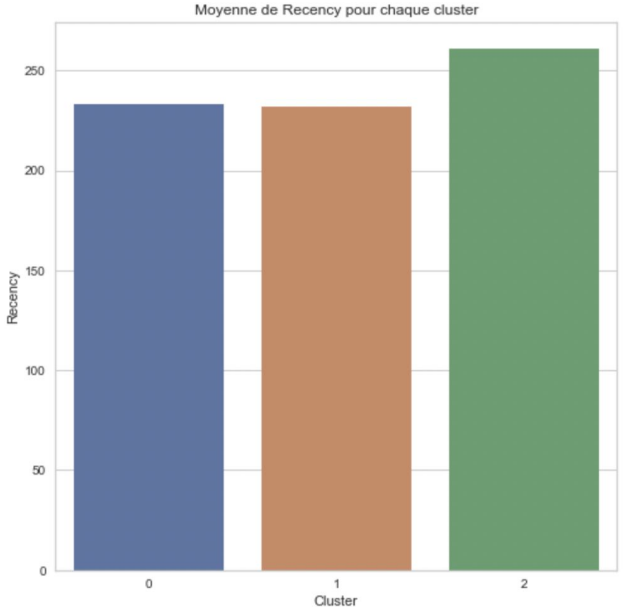


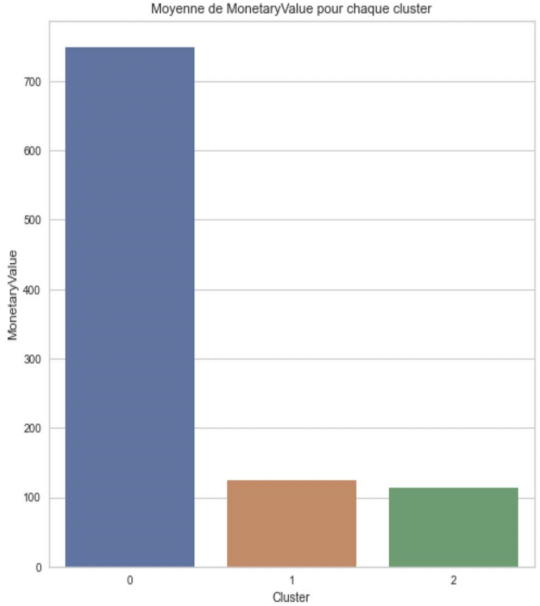




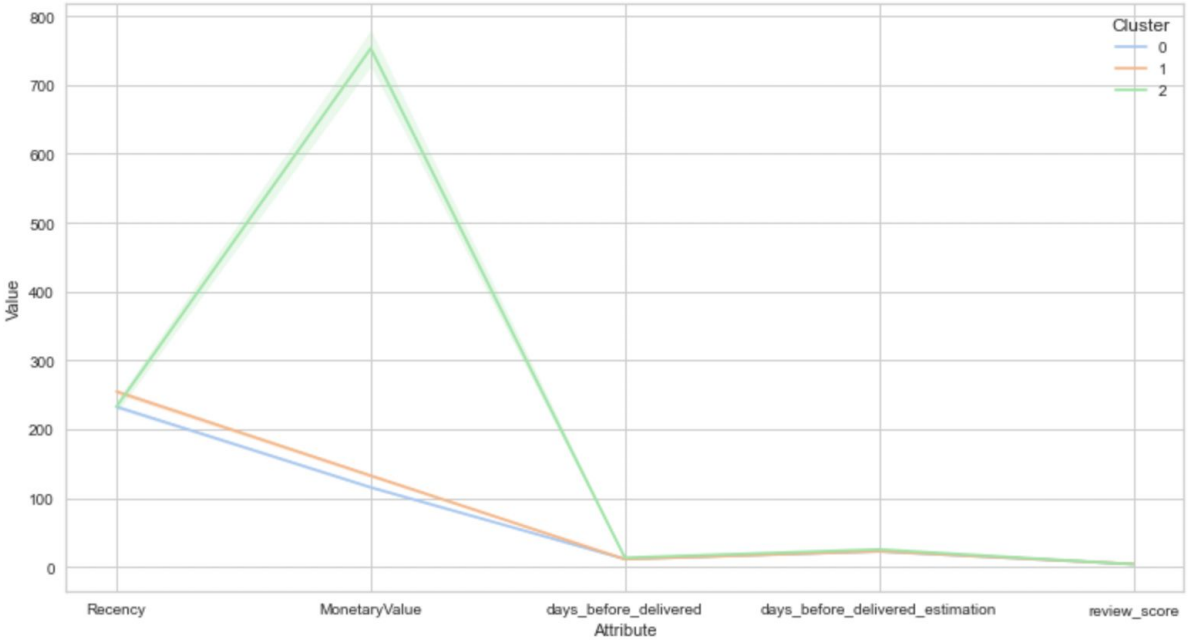
la moyenne des variables par cluster CAH







Visualisation parallel plot des valeurs moyennes par cluster CAH



à cause du diminuer la taille de notre données(sample 30% à cause de problème de Ram)on ne peut pas bien nommer des segments, mais on voit que :

- le segment 0 :les clients gros acheteurs qui ont acheté les paniers normaux

Conclusion sur le choix du meilleur algo

Pour comparer les algos on a 2 critères :

1 : le silhouette score

- le silhouette score Kmeans: 0.34
- le silhouette score DBscan : 0.038
- le silhouette score Agglomerative Hierarchical Clustering : 0.81 (Il est un bon score puisqu'on a fait algo sur 30% de données)

2 : Critère de ré-utilisation : Pour faire des segments sur des nouveaux clients, est-il possible d'appliquer le modèle sur de nouvelles données ?

- Kmeans est une méthode réutilisable sur les nouvelles données donc on peut utiliser Kmeans pour prédire des segments.
- Dbscan et CH utilisent les données qu'on avait avant pour prédire donc c'est plus compliqué de faire prédiction avec ces deux méthodes

Puisque il y a pas de nuages qui se forme distangutement il y a pas des groupes qui se détache :

- Kmeans est plus intéressant parce qu'on a pas une typologie particulière de l'observation ils sont plutôt dégroupés .
- DBscan il va être hyper compliqué la densité est plus constante uniforme entre tous les points.

Notre choix:

- Kmeans (réutilisable) ⇒
- CH

Ok

Stabilité et maintenance

On voudrait fournir une description actionable pour une utilisation optimale.

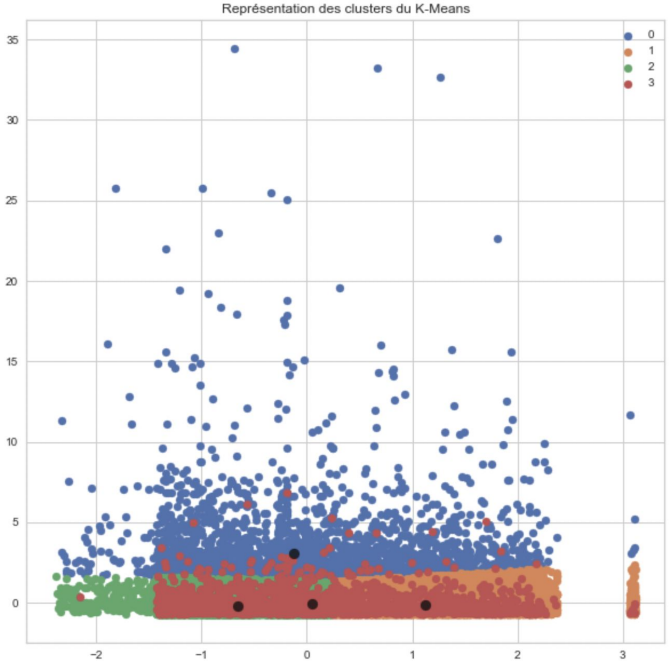
une proposition de contrat de maintenance basée sur une analyse de la stabilité des segments au cours du temps.

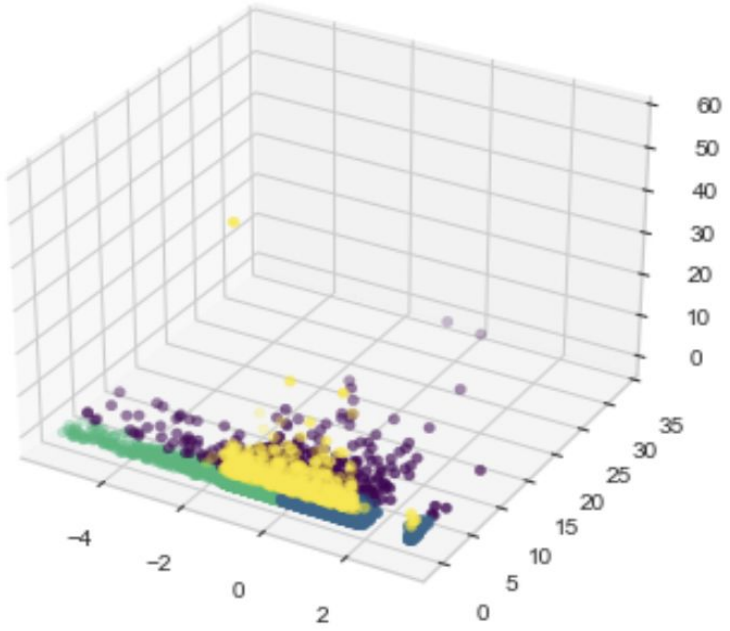
Donc on veut savoir si notre algorithme fournira les mêmes segmentations dans le futur, alors on va simuler le futur.

- On crée un dataset :
  - df0 qui ne prend en compte que les 12 premiers mois de la période d'étude de 2016_10 à 2017_10
  - df1 qui prend en compte df0 + 1 mois 2017_11
  - df2 qui prend en compte df1 + 1 mois 2017_12
  - df3 qui prend en compte df2 + 1 mois 2018_01
  - df4 qui prend en compte df3 + 1 mois 2018_02
  - df5 qui prend en compte df4 + 1 mois 2018_03
  - df6 qui prend en compte df5 + 1 mois 2018_04
  - df7 qui prend en compte df6 + 1 mois 2018_05
  - df8 qui prend en compte df7 + 1 mois 2018_06
  - df9 qui prend en compte df8 + 1 mois 2018_07
  - df10 qui prend en compte df9 + 1 mois 2018_08

- Apprendre tous les clusters sur les bases artificielles *B1*, *B2*, Bn -> *C1 C2 C3* (les clustering futures), pour tous les B1 à 10
- Segmenter *B1 B2 B3* avec *C0 pour tous les B de 1 à 10*
- Segmenter *B1 B2 B3* avec les clustering respectifs *C1 C2 etc* pour tous les B et C de 1 à 10

```
c0_cluster.fit(b0 data)
c1 cluster.fit(b1 data)
b1_by_c0 = c0_cluster.predict(b1_data)
b1_by_c1 = c1_cluster.predict(b1_data) b2_by_c2 = 2_cluster.predict(b2_data)
```


Représentation des clusters du K-Means
0
1
2
3
35
30
25
20
15
10
5
0
−2
−1
0
1
2
3

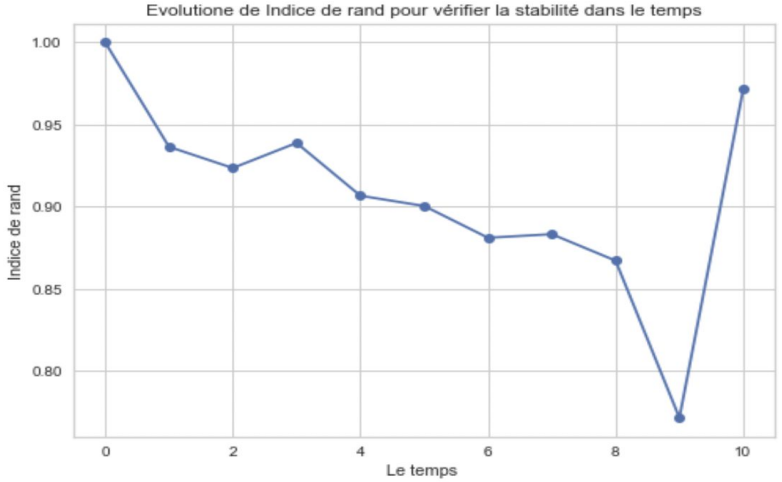

60
50
40
30
20
10
0
35
30
25
20
15
10
5
0
−4
−2
0
2

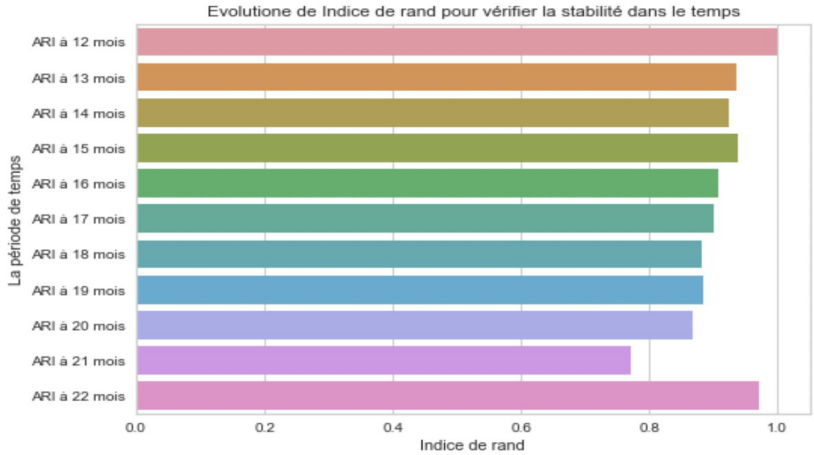
l'indice de rand

- On compare la segmentation de *B1 B2 B3* entre *C0* (livré au client) et les autres (*C1* etc) avec l'indice de rand etc pour tous les B de 1 à 10

```
rand(b1_by_c0, b1_by_c1)
rand(b2_by_c0, b2_by_c2)
```

	ARI par mois	Indice de rand
0	ARI à 12 mois	1.000000
1	ARI à 13 mois	0.936466
2	ARI à 14 mois	0.923548
3	ARI à 15 mois	0.938822
4	ARI à 16 mois	0.906692
5	ARI à 17 mois	0.900460
6	ARI à 18 mois	0.881121
7	ARI à 19 mois	0.883310
8	ARI à 20 mois	0.867192
9	ARI à 21 mois	0.771611
10	ARI à 22 mois	0.971389





Conclusion

Conclusion:

- Mise en application des algorithmes de classification non supervisée et application à un problème métier
- Kmeans et CAH sont deux algorithmes qu'on peut utiliser mais Kmeans est plus adapté pour faire de la prédiction sur de nouvelles données
- la période de maintenance: 6 mois

Améliorations:

Opportunités d'amélioration du clustering:

- Définir les features les plus intéressantes avec le client
- Utilisation des caractéristiques du produit (catégorie, matériaux, etc)
- Données plus précises sur les clients comme sexe, âge