olist

Segmenter des clients d'un site e-commerce

Sommaire

olist

olist

1. Contexte du projet

Contexte du projet

Problématique : L'équipe e-commerce de l'entreprise Olist souhaite une solution de segmentation des clients pour leurs campagnes de communications.

Objectif : fournir à l'équipe Marketing une description actionnable d'une segmentation client.

Missions :

- Nettoyer et analyser les données.
- Utiliser des méthodes non supervisées pour regrouper les clients similaires.
- Mettre en place une simulation de maintenance pour déterminer la fréquence de mise à jour du modèle de segmentation.

Résultat attendu : 3 notebooks :

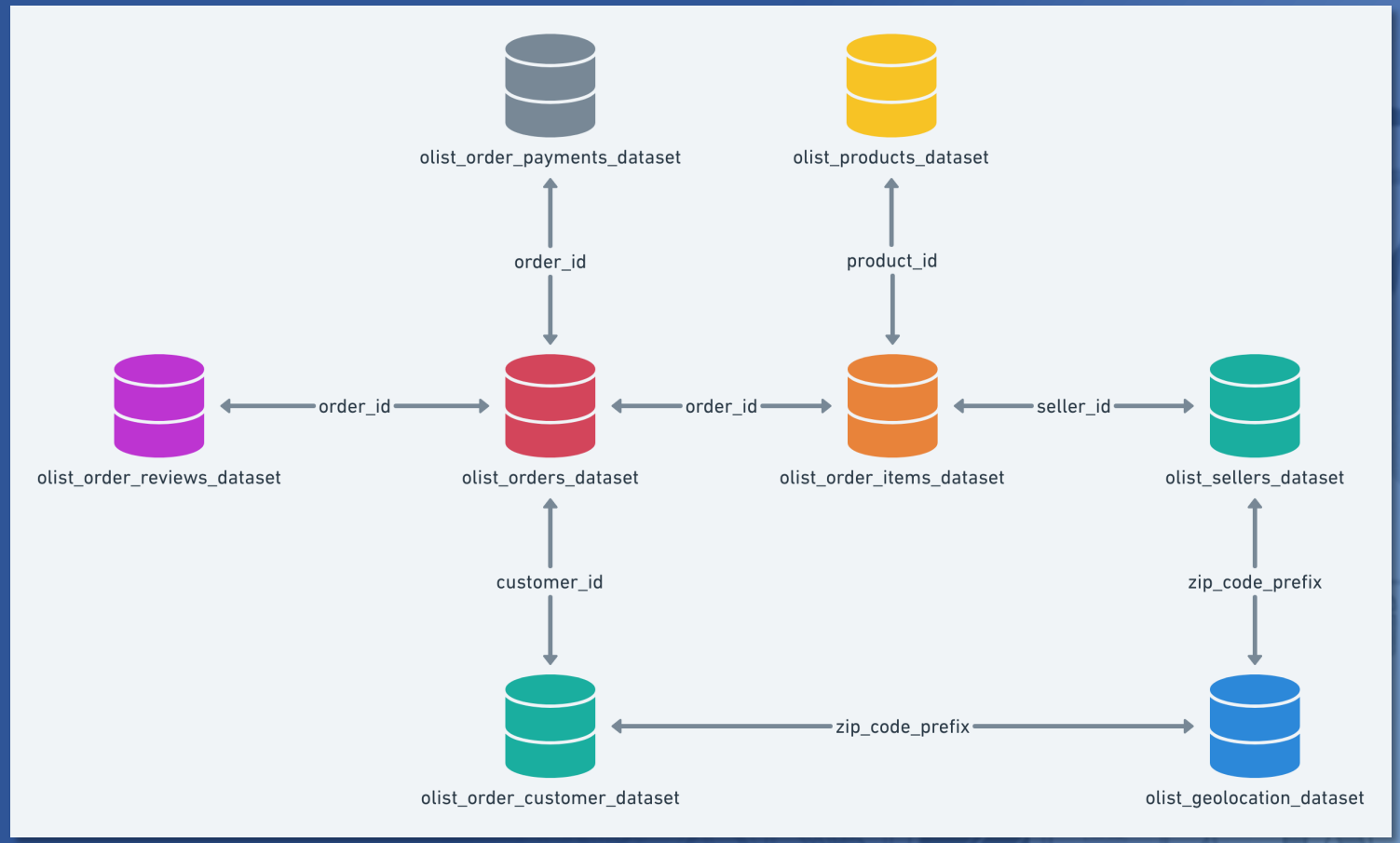
- Analyse exploratoire des données,
- Essais des différentes approches de modélisation,
- Simulation pour déterminer la fréquence nécessaire de mise à jour.

olist

olist

2. Présentation, nettoyage et analyse des données

Présentation du jeu de données




```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115299 entries, 0 to 115298
Data columns (total 43 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   order_id                       115299 non-null  object
 1   customer_id                    115299 non-null  object
 2   order_status                   115299 non-null  object
 3   order_purchase_timestamp       115299 non-null  object
 4   order_approved_at              115285 non-null  object
 5   order_delivered_carrier_date   114111 non-null  object
 6   order_delivered_customer_date  112912 non-null  object
 7   order_estimated_delivery_date  115299 non-null  object
 8   order_item_id                  115299 non-null  int64
 9   product_id                     115299 non-null  object
 10  seller_id                      115299 non-null  object
 11  shipping_limit_date            115299 non-null  object
 12  price                          115299 non-null  float64
 13  freight_value                  115299 non-null  float64
 14  product_category_name          115299 non-null  object
 15  product_name_lenght            115299 non-null  float64
 16  product_description_lenght     115299 non-null  float64
 17  product_photos_qty             115299 non-null  float64
 18  product_weight_g               115298 non-null  float64
 19  product_length_cm              115298 non-null  float64
 20  product_height_cm              115298 non-null  float64
 21  product_width_cm               115298 non-null  float64
 22  seller_zip_code_prefix         115299 non-null  int64
 23  seller_city                    115299 non-null  object
 24  seller_state                   115299 non-null  object
 25  payment_sequential             115299 non-null  int64
 26  payment_type                   115299 non-null  object
 27  payment_installments           115299 non-null  int64
 28  payment_value                  115299 non-null  float64
 29  review_id                      115299 non-null  object
 30  review_score                   115299 non-null  int64
 31  review_comment_title           13756 non-null   object
 32  review_comment_message         48780 non-null   object
 33  review_creation_date           115299 non-null  object
 34  review_answer_timestamp        115299 non-null  object
 35  customer_unique_id             115299 non-null  object
 36  customer_zip_code_prefix       115299 non-null  int64
 37  customer_city                  115299 non-null  object
 38  customer_state                 115299 non-null  object
 39  geolocation_lat                115299 non-null  float64
 40  geolocation_lng                115299 non-null  float64
 41  geolocation_city               115299 non-null  object
 42  geolocation_state              115299 non-null  object
dtypes: float64(12), int64(6), object(25)
memory usage: 37.8+ MB
```

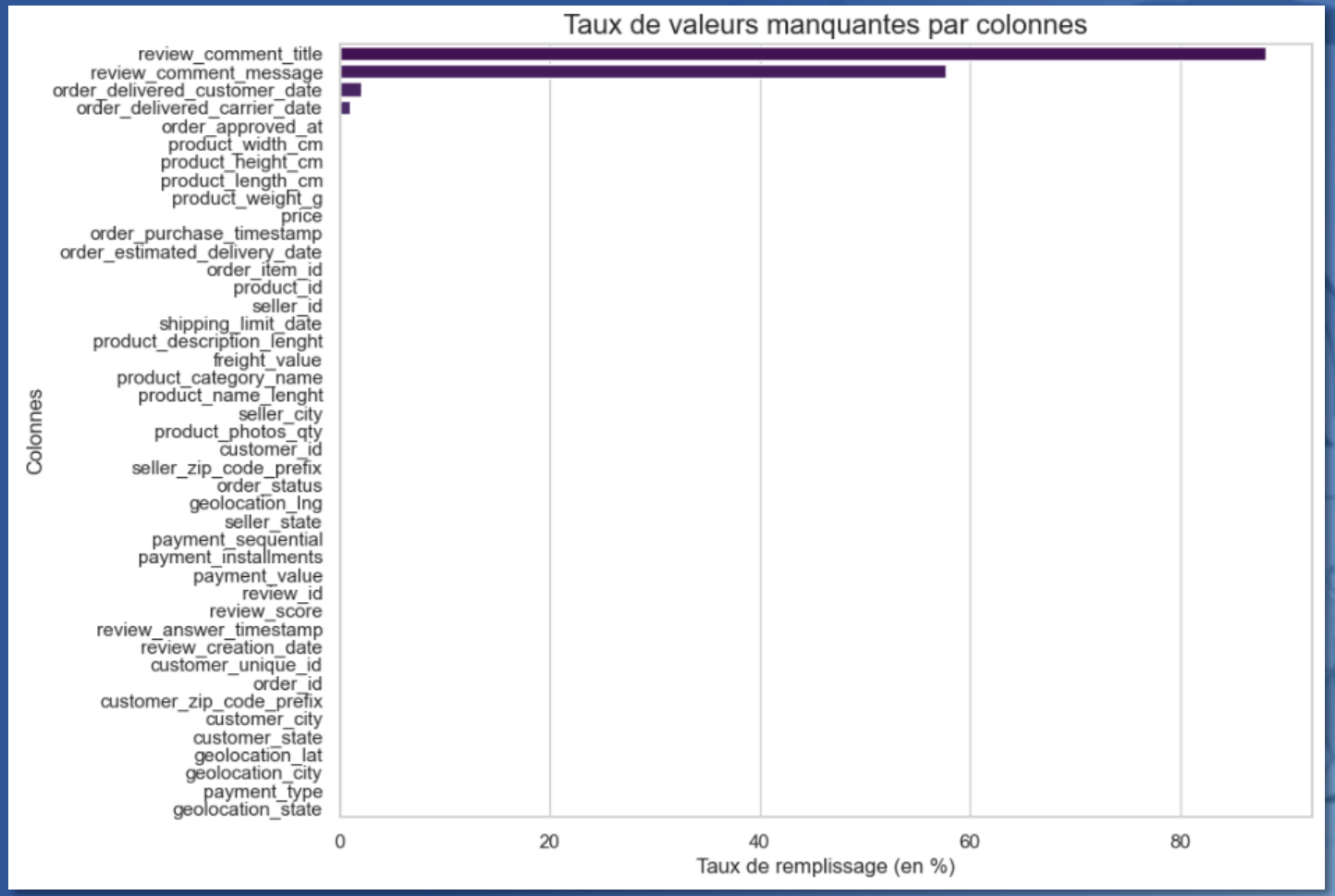
Une fois les jointures réalisées, les données correspondent à toutes les commandes passées.

Les individus ne sont donc pas les clients mais leurs commandes. Plusieurs lignes peuvent donc concerner un seul et même client.

olist

Nettoyage des données .1/3

Valeurs manquantes



Suppression des variables de commentaires et des titres de commentaires.

Trop de valeurs manquantes, peu exploitable et pas utilisable dans le cadre d'une segmentation clients.

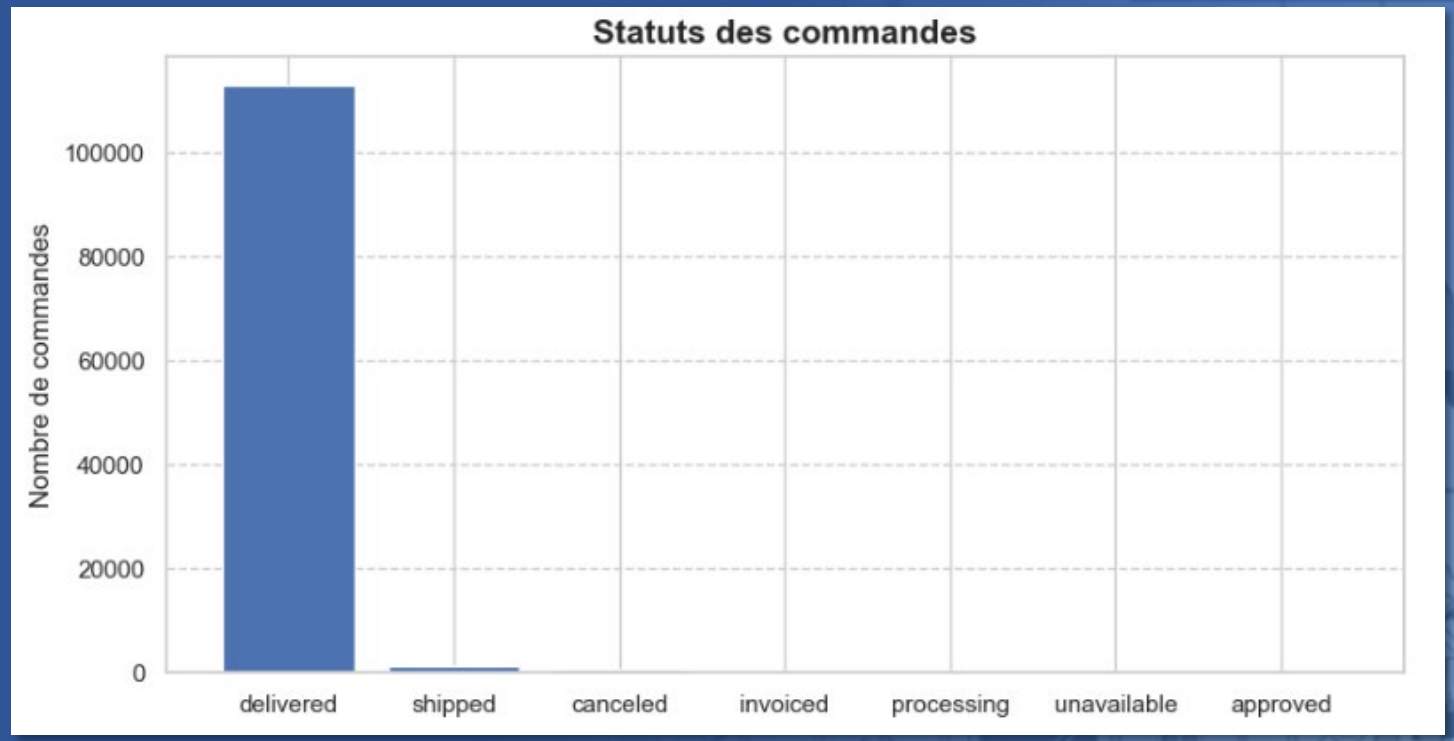
Nombre de variables supprimées : 2

Nombre de variables restantes après action : 41

olist

Nettoyage des données .2/3

Statuts des commandes



Suppression des commandes autres que les commandes délivrées.

Résultats toutes ont le même statut, on peut supprimer la variable.

Nombre de variables supprimées : 1
Nombre de variables restantes après action : 40

olist

Nettoyage des données .3/3

Suppression des colonnes non-utiles à la segmentation des clients

```
colonnes_à_supprimer = ['customer_id', 'order_approved_at', 'order_delivered_carrier_date',
                        'order_delivered_customer_date', 'order_estimated_delivery_date', 'order_item_id',
                        'shipping_limit_date', 'product_name_lenght', 'product_description_lenght',
                        'product_photos_qty', 'product_weight_g', 'product_length_cm', 'product_height_cm',
                        'product_width_cm', 'seller_zip_code_prefix', 'seller_city', 'seller_state',
                        'payment_sequential', 'payment_type', 'payment_installments', 'review_id', 'review_creation_date',
                        'review_answer_timestamp']

df_merge = df_merge.drop(columns=colonnes_à_supprimer).reset_index(drop=True)
```

Suppression des variables qui ne caractérisent pas les clients et donc qui ne seront pas utiles à la segmentation.

Data Set après suppression :

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112913 entries, 0 to 112912
Data columns (total 17 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   order_id                 112913 non-null  object
 1   order_purchase_timestamp 112913 non-null  object
 2   product_id               112913 non-null  object
 3   seller_id                112913 non-null  object
 4   price                    112913 non-null  float64
 5   freight_value            112913 non-null  float64
 6   product_category_name    112913 non-null  object
 7   payment_value            112913 non-null  float64
 8   review_score             112913 non-null  int64
 9   customer_unique_id       112913 non-null  object
 10  customer_zip_code_prefix 112913 non-null  object
 11  customer_city            112913 non-null  object
 12  customer_state           112913 non-null  object
 13  geolocation_lat          112913 non-null  float64
 14  geolocation_lng          112913 non-null  float64
 15  geolocation_city         112913 non-null  object
 16  geolocation_state        112913 non-null  object
dtypes: float64(5), int64(1), object(11)
memory usage: 14.6+ MB
```

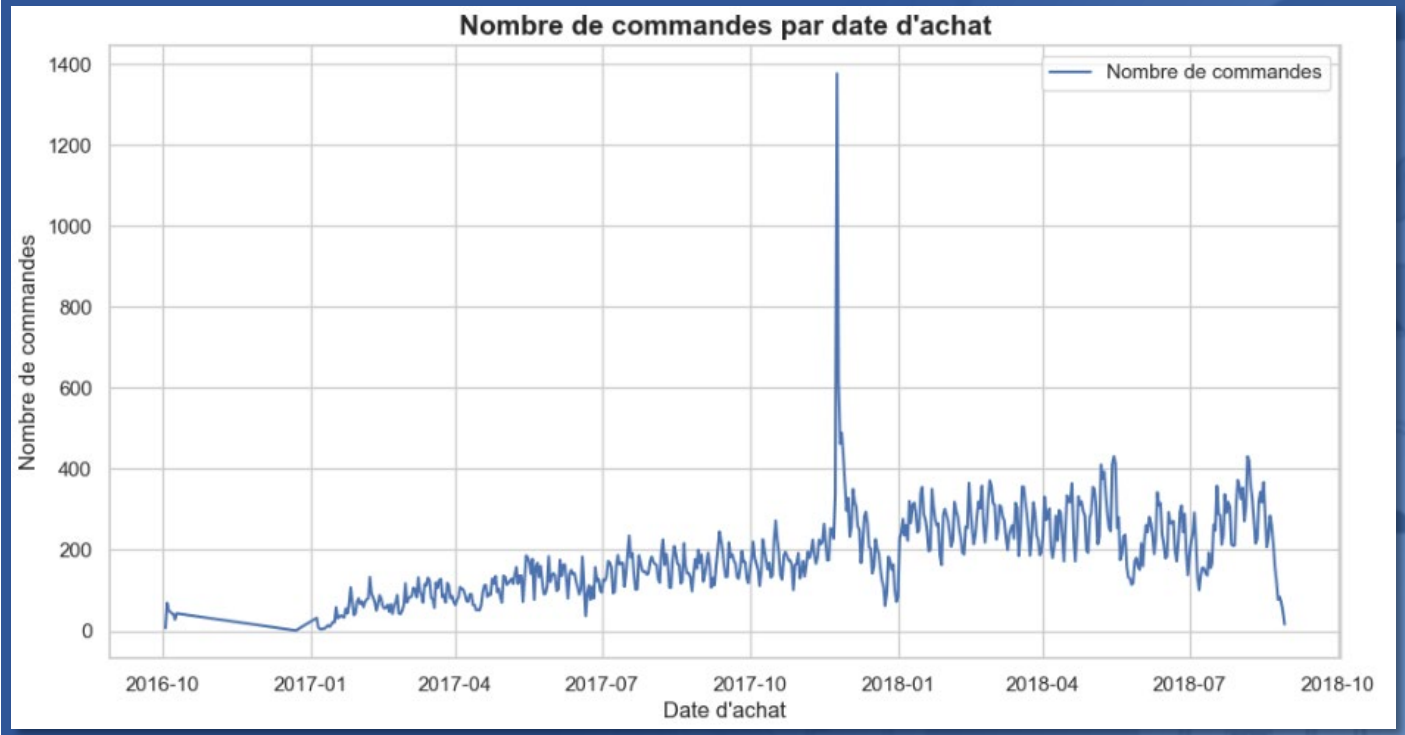
Nombre de variables supprimées : 23

Nombre de variables restantes après action : 17

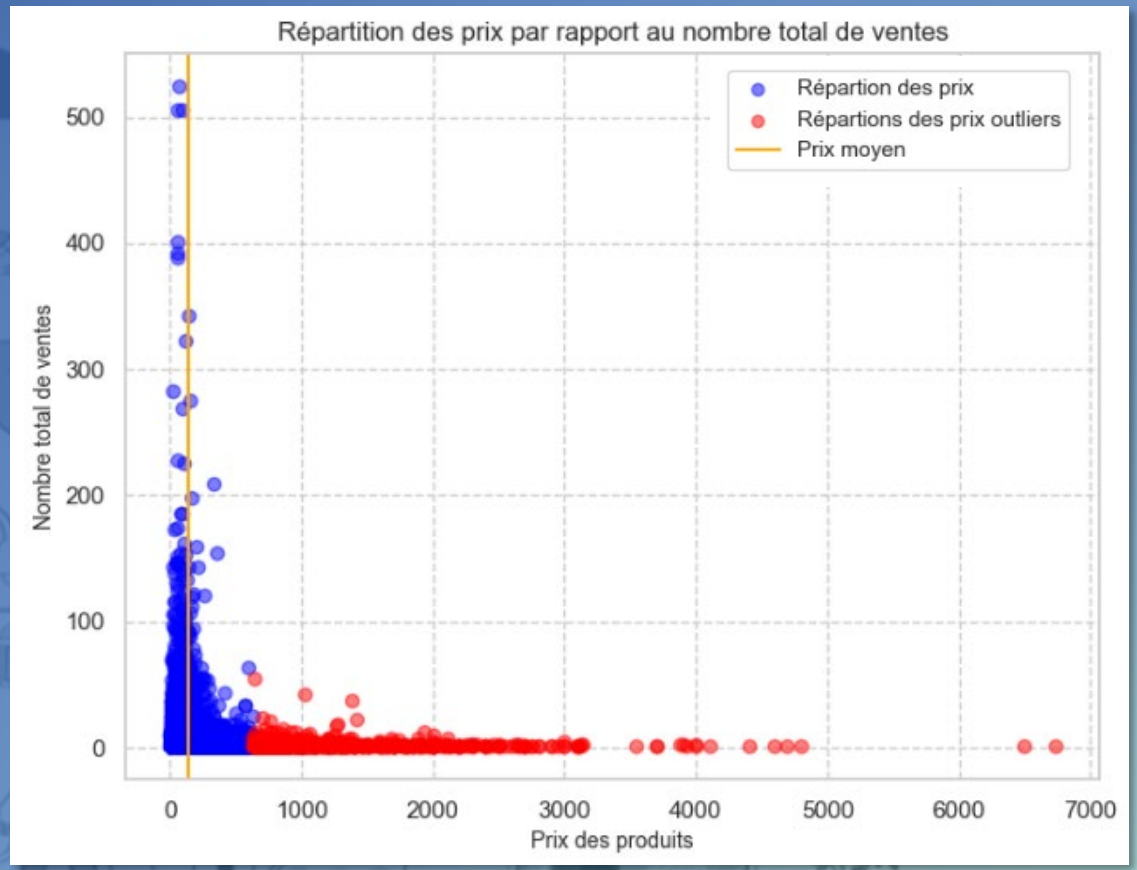
olist

Analyse Exploratoire .1/5

Nombre de commandes par dates d'achat et répartition des prix par rapport au nombre total de ventes



On peut voir que le nombre de commande en fonction de la date d'achat est plutôt bien réparti.
A noter qu'il y a une forte hausse au moment des fêtes de fin d'années (2017-12).

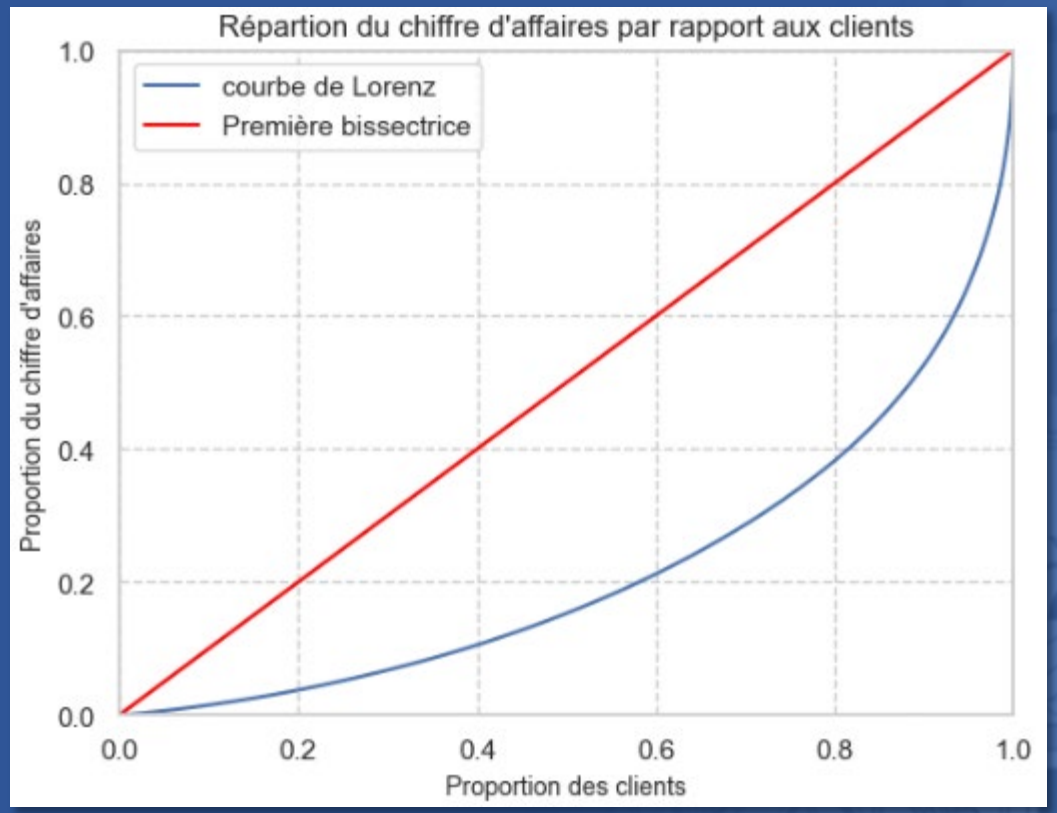


Les produits les plus vendus sont les produits « moins chers », plus le prix des produits augmentent et moins ils sont vendus.

olist

Analyse Exploratoire .2/5

Profil des clients



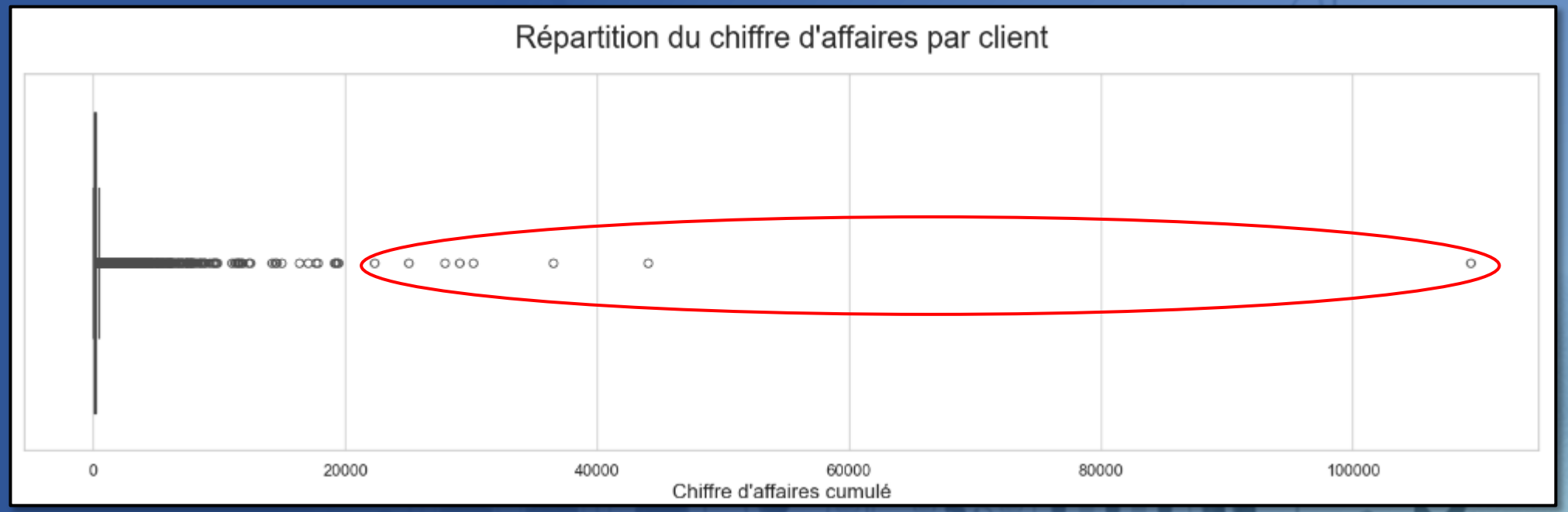

```
L'indice de Gini est de : 0.56
```

Un indice de Gini de 0.56 veut dire que la répartition du chiffre d'affaires entre les clients est assez inégale. Certains clients génèrent une part plus importantes du chiffre d'affaires.

olist

Analyse Exploratoire .3/5

Répartition du chiffre d'affaires par clients



Nous avons des clients avec un très fort chiffre d'affaires cumulé. Il pourrait s'agir d'erreurs ou de clients professionnels.

La solution est donc de supprimer ces clients « atypiques » pour ne pas biaiser nos segmentations futures.

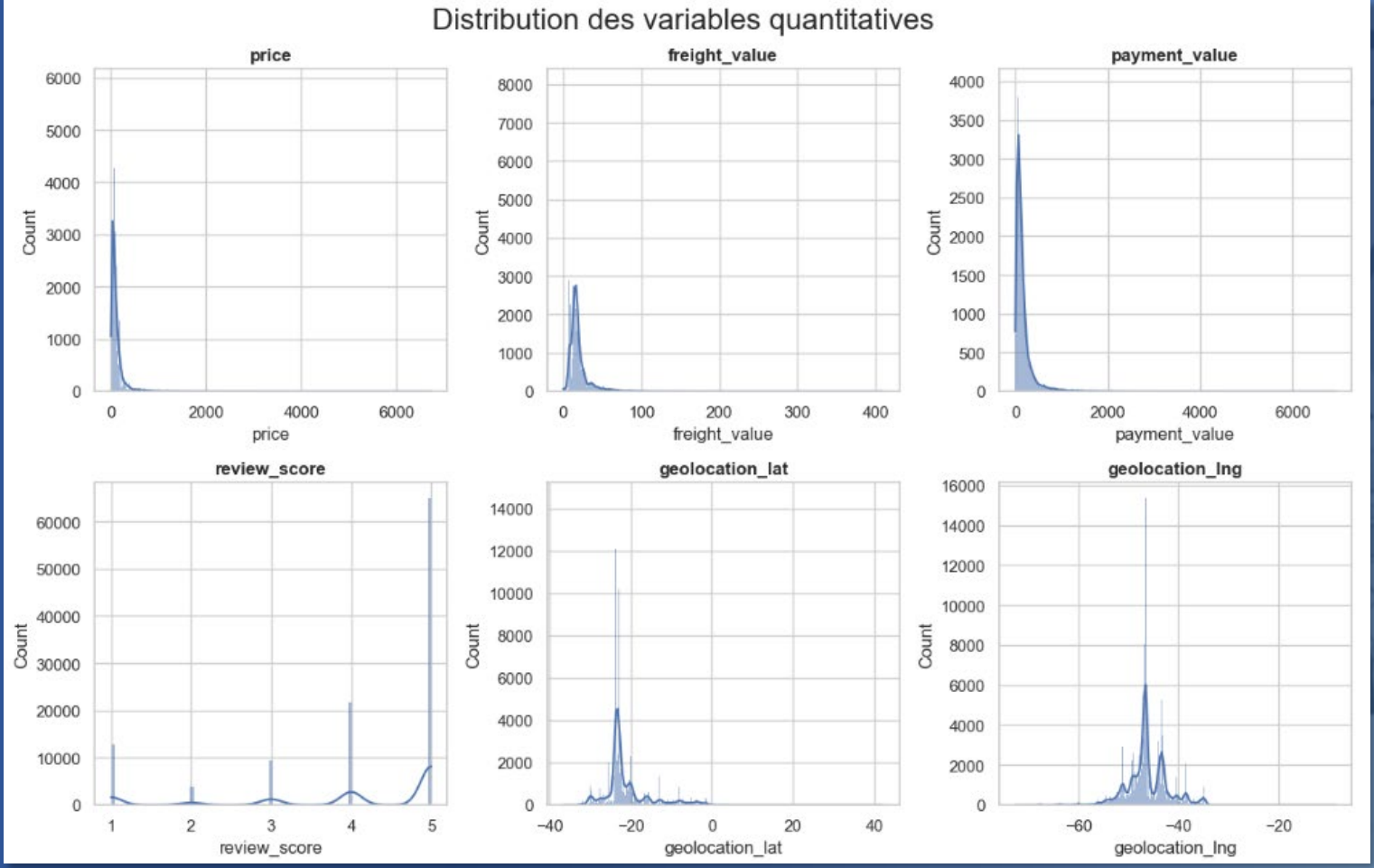
Nous allons donc supprimer les données des 8 clients ayant un chiffre d'affaires cumulé supérieur à 20 000.

Ces clients doivent donc être traités à part.

olist

Analyse Exploratoire .4/5

Distribution des variables quantitatives



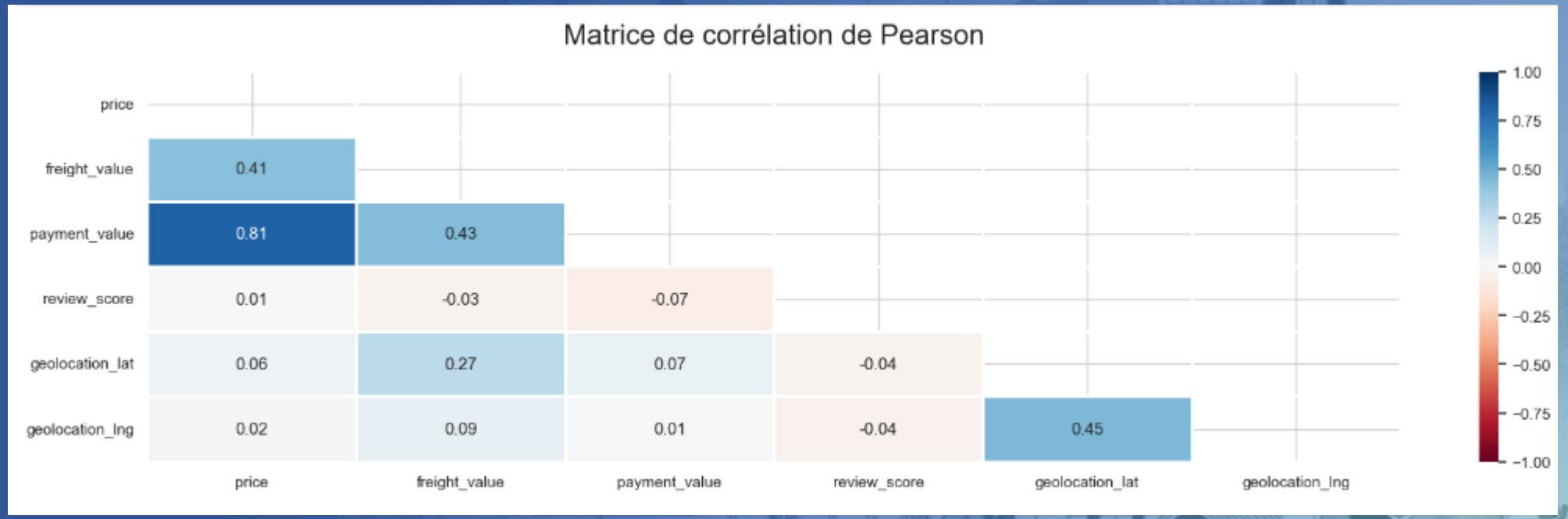
Les variables ne suivent pas une loi normale.

Il faudra normaliser les données avant l'entraînement des modèles de segmentation.

olist

Analyse Exploratoire .5/5

Analyse des corrélations (analyse bivariée)



olist

3. Feature engineering

Feature Engineering .1/2

Variables RFM (Récence, Fréquence et Montant)

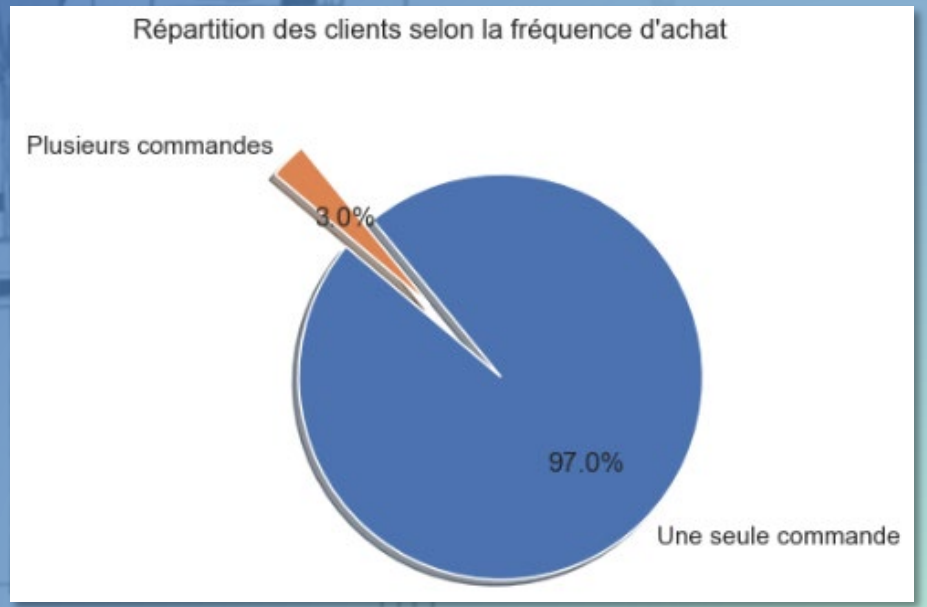
A partir de maintenant, il faut attribuer les données à chacun des clients uniques.

En effet, chaque ligne de notre jeu de donné correspond à une seule et unique commande.
Le but est donc d'agréger les données sur chaque client unique.

R La récence est calculée en soustrayant la dernière date de commande connue du Dataset et la dernière date de commande du client.

F La fréquence est calculée en comptant le nombre de commandes passées pour chaque clients.

M Le montant est calculée en additionnant le montant des commandes pour chaque clients.



olist

Feature Engineering .2/2

Moyenne des review scores

Puis, la moyenne des scores laissés par les clients a été rajoutée.

Le Dataset final avec les données des clients comporte donc 91 225 individus pour 5 variables.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91225 entries, 0 to 91224
Data columns (total 5 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   customer_unique_id   91225 non-null  object 
 1   R_recence            91225 non-null  int64  
 2   F_frequence          91225 non-null  int64  
 3   M_montant            91225 non-null  float64
 4   Mean_review_score    91225 non-null  float64
dtypes: float64(2), int64(2), object(1)
memory usage: 3.5+ MB
```

	R_recence	F_frequence	M_montant	Mean_review_score
count	91225.000000	91225.000000	91225.000000	91225.000000
mean	236.149685	1.032809	208.804403	4.154727
std	152.612152	0.205937	440.161224	1.279109
min	0.000000	1.000000	9.590000	1.000000
25%	113.000000	1.000000	63.830000	4.000000
50%	217.000000	1.000000	112.870000	5.000000
75%	344.000000	1.000000	202.740000	5.000000
max	694.000000	14.000000	19457.040000	5.000000

olist

olist

4. Mise en place des algorithmes de clustering

Kmeans avec les données RFM .1/3

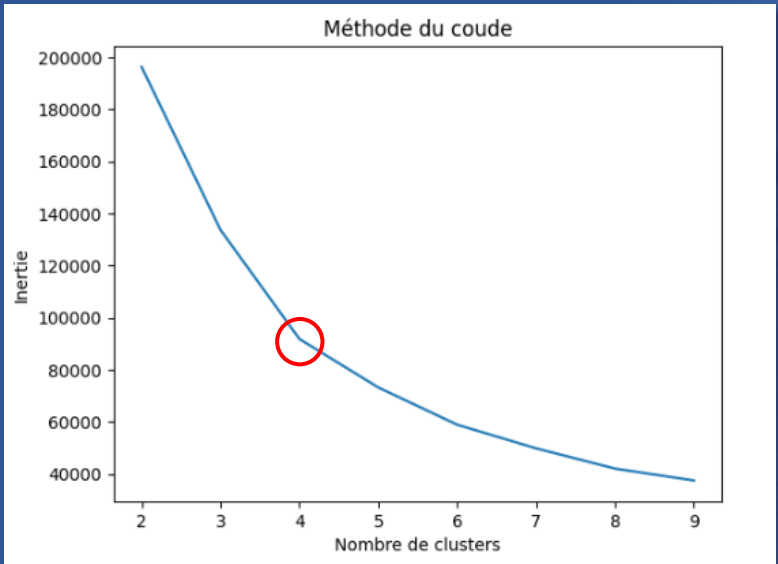
Normalisation des données et choix du nombre optimal de clusters

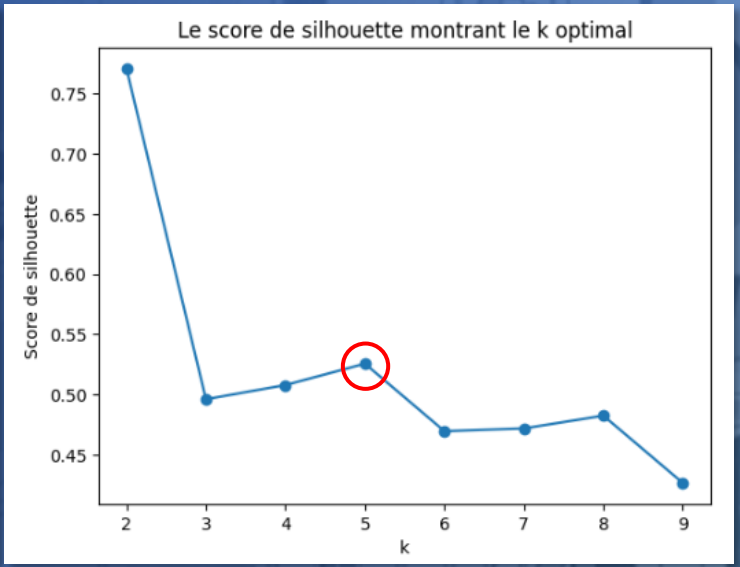
Normalisation des données :

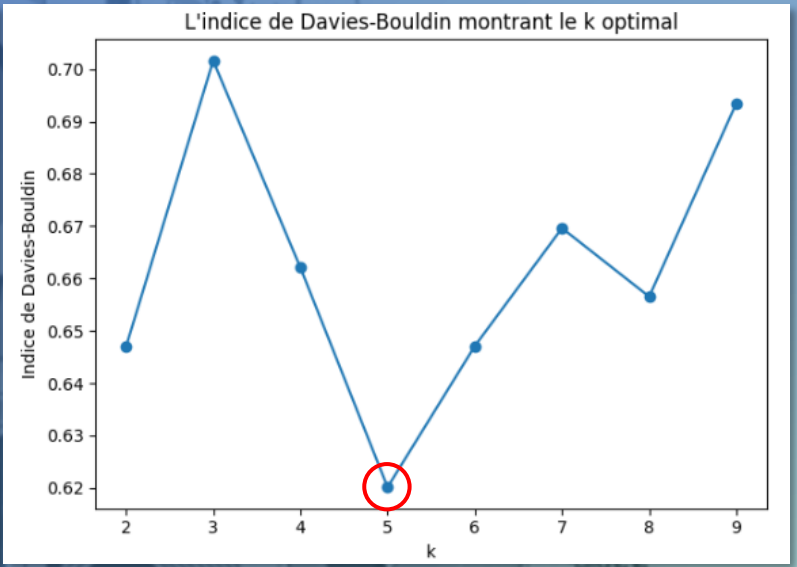
```
# Normaliser les données
scaler = StandardScaler()
df_normalized = scaler.fit_transform(df_RFM)
```

Afin de normaliser les données, un StandardScaler a été utilisé pour rendre les variables indépendantes de leur unité et de leur échelle d'origine.

Choix du nombre optimal de clusters :







	Type	n_clusters	Silhouette	Davies_Bouldin
0	KMeans	2	0.771068	0.646928
3	KMeans	5	0.525578	0.620122
2	KMeans	4	0.507793	0.662084
1	KMeans	3	0.495990	0.701606

En utilisant la méthode du coude, nous constatons que le nombre idéal de clusters se situe entre 4 et 5.

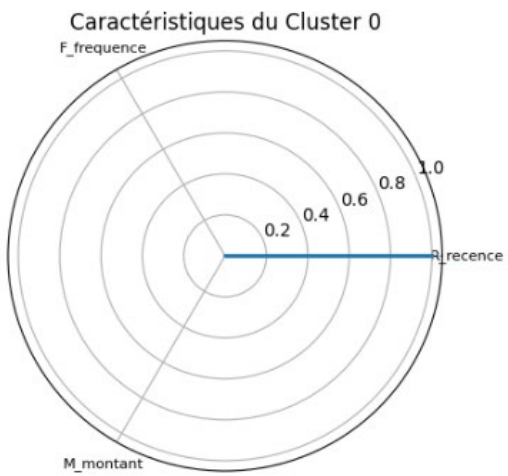
En se basant sur le score Silhouette, les options de 4 ou 5 clusters sont viables, et on peut voir ensuite que l'indice de Davies-Bouldin est minimisé pour 5 clusters.

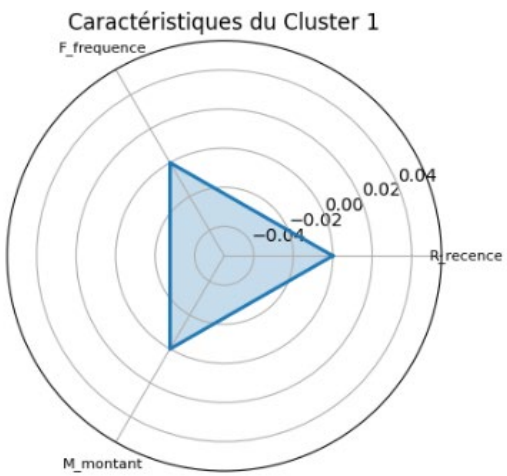
La meilleure option est donc 5 clusters.

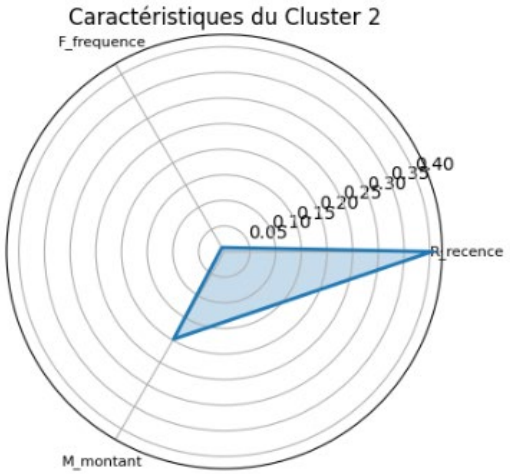
olist

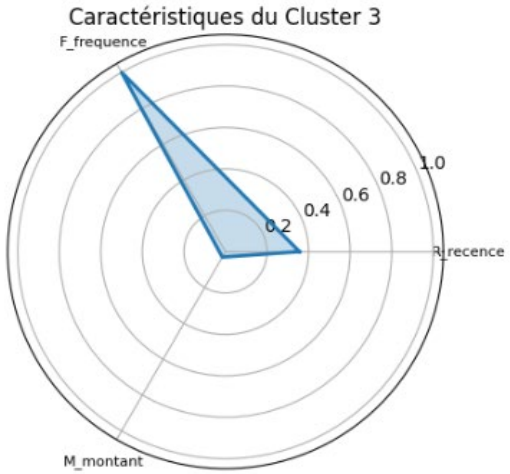
Kmeans avec les données RFM .2/3

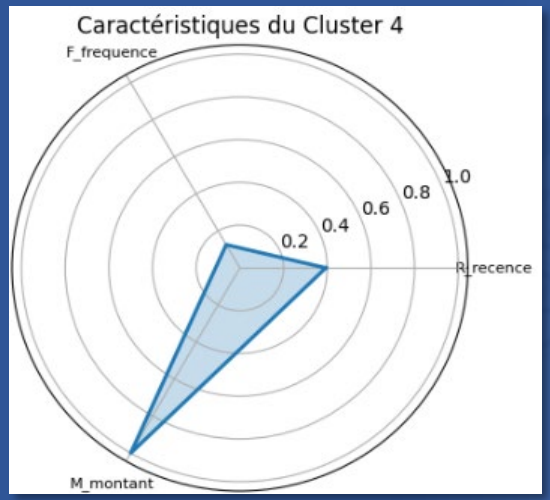
Interprétation des clusters











Centroïds du Kmeans :

KMeans_label	R_recence	F_frequence	M_montant
0	386.256522	1.000000	157.379171
1	126.370664	1.000000	156.304528
2	231.435644	1.010396	1714.466198
3	218.953777	2.110485	396.419410
4	228.565574	1.139344	8053.073033

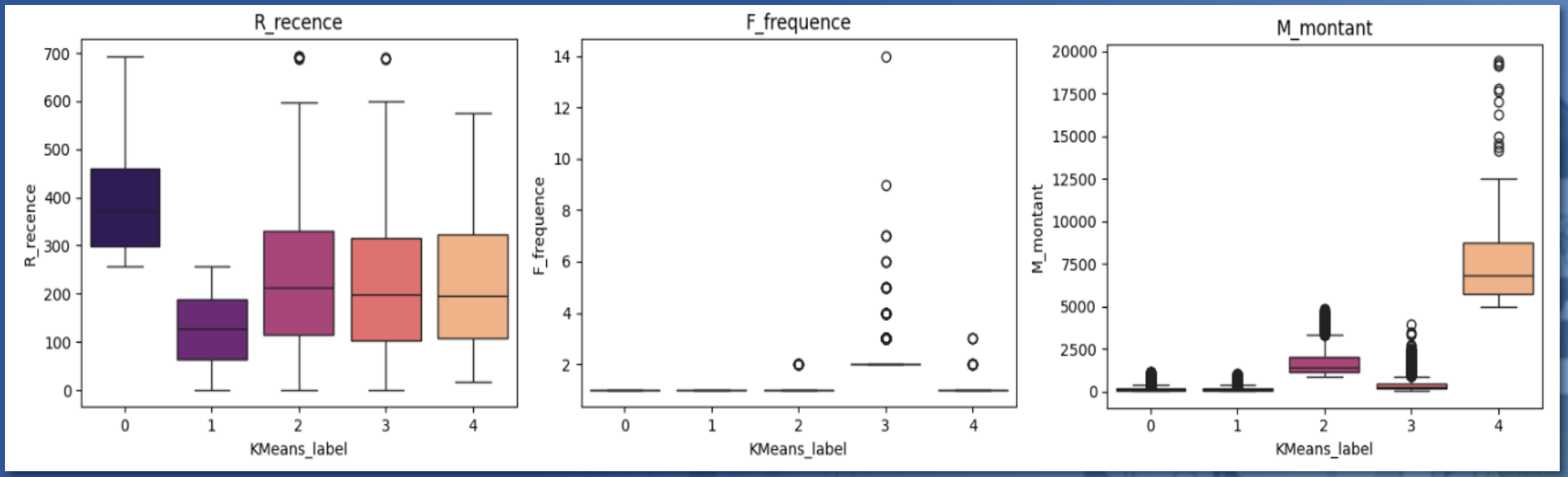
Nombre de client par clusters :

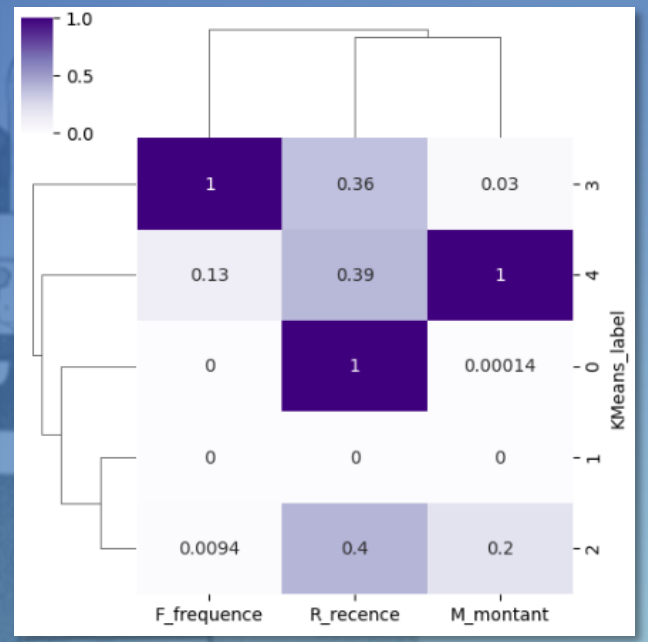
KMeans_label	count
1	49700
0	36722
3	2661
2	2020
4	122

olist

Kmeans avec les données RFM .3/3

Interprétation des clusters





Interprétation métier des clusters :

Le **cluster 0** correspond aux *clients perdus*. Ils ont commandé une seule fois pour un montant modéré il y a 386 jours en moyenne.

Le **cluster 1** correspond aux derniers clients ayant commandé ; ils peuvent être assimilés à de *nouveaux clients* car c'était leur première commande.

Le **cluster 2** correspond aux *clients ayant dépensé une somme importante sur le site* ; dans la majorité des cas ces clients n'ont commandé qu'une seule fois.

Le **cluster 3** correspond aux *clients fidèles* ; ces clients ont commandé au moins 2 fois sur le site.

Le **cluster 4** correspond aux *clients ayant dépensé le plus sur le site* ; dans la majorité des cas ces clients n'ont commandés qu'une seule fois.

olist

CAH avec les données RFM .1/3

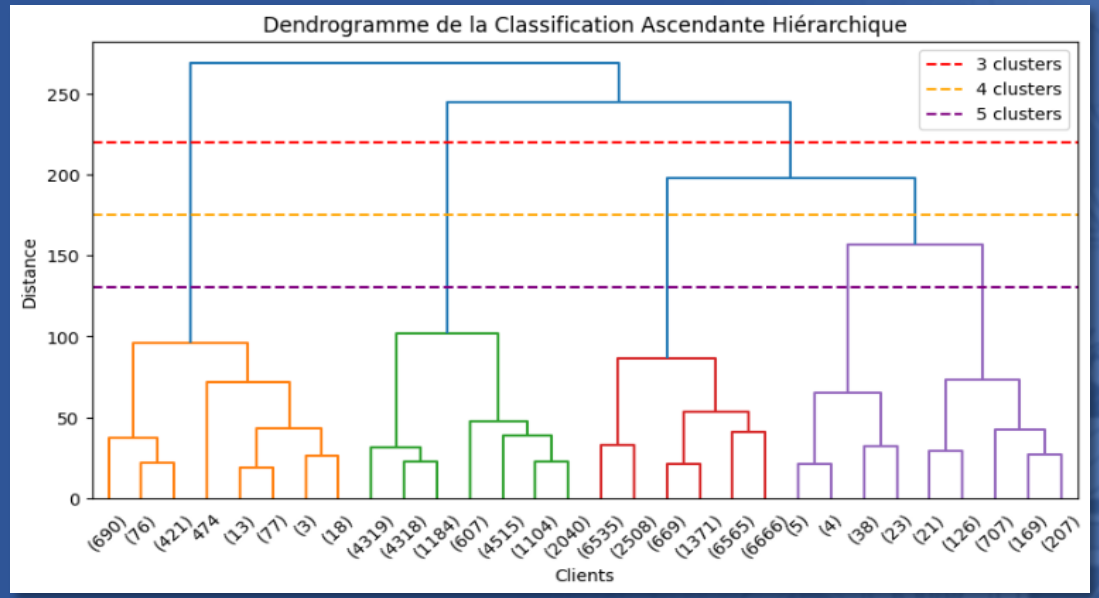
Échantillonnage du Data Set et choix du nombre optimal de clusters

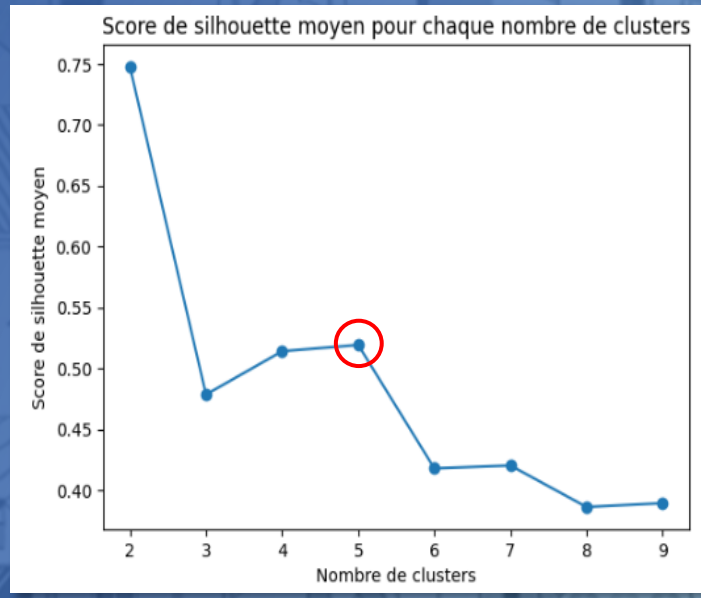
Échantillonnage des données :

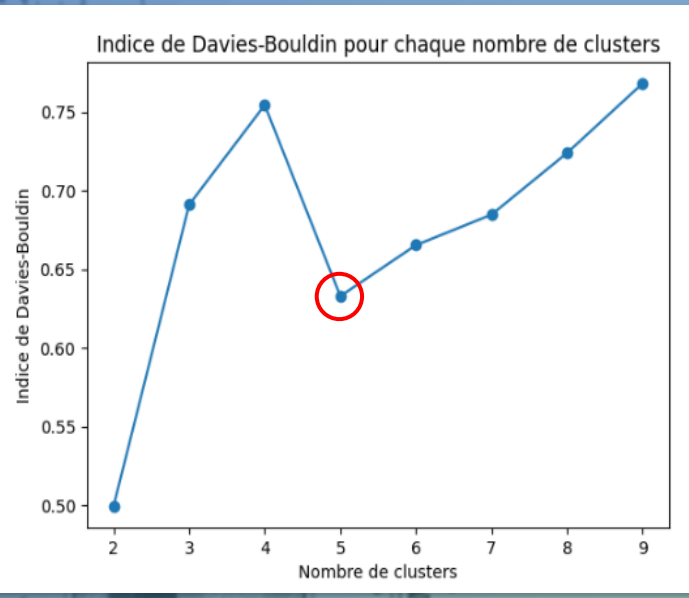
```
# Le nombre de données est trop volumineux (cela donne une erreur de type 'MemoryError')
# Prennons un échantillon
échantillon = df_RFM.sample(n=45000, random_state=0)
```

Afin d'éviter les erreurs dû à la mémoire vive (« MemoryError »), il a été nécessaire d'échantillonner le Data Set et de ne prendre que 45 000 clients.

Choix du nombre optimal de clusters :







	Type	n_clusters	Silhouette	Davies_Bouldin
0	CAH	2	0.747728	0.499237
3	CAH	5	0.519143	0.632865
2	CAH	4	0.513991	0.754483
1	CAH	3	0.478432	0.691042

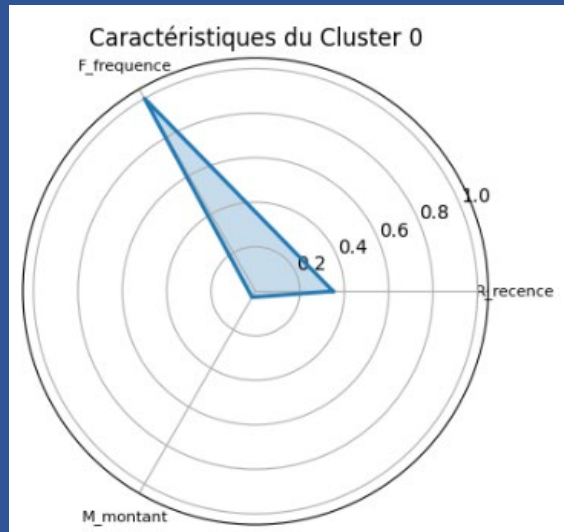
En utilisant le dendrogramme, on pourrait opter pour 3 options : 3, 4 ou 5 clusters.

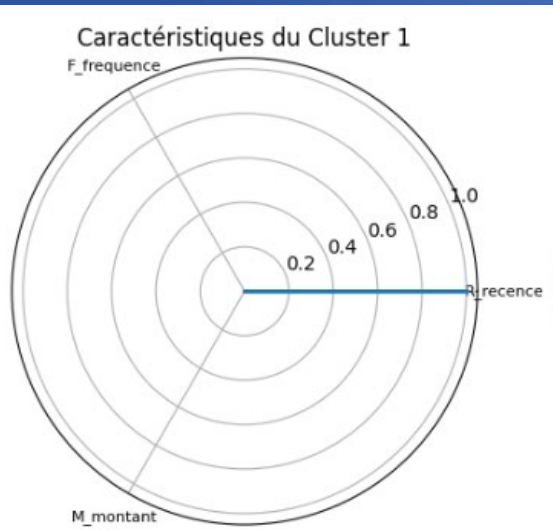
En se basant sur le score Silhouette et l'indice de Davies-Bouldin on optera pour 5 clusters.

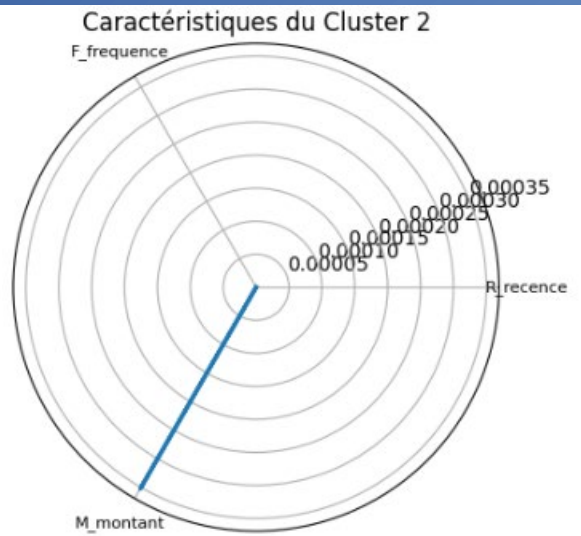
olist

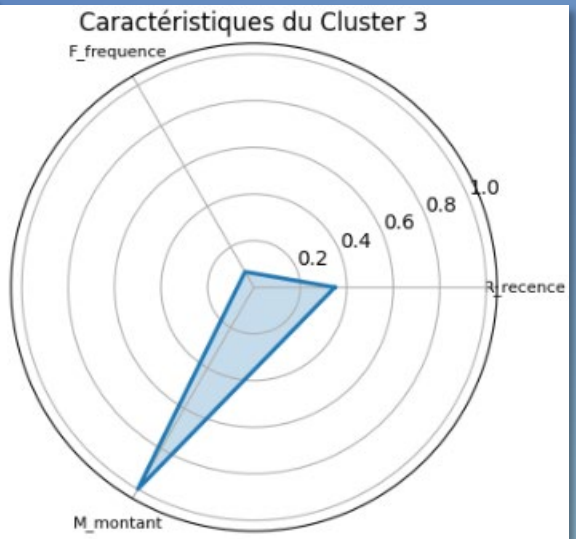
CAH avec les données RFM .2/3

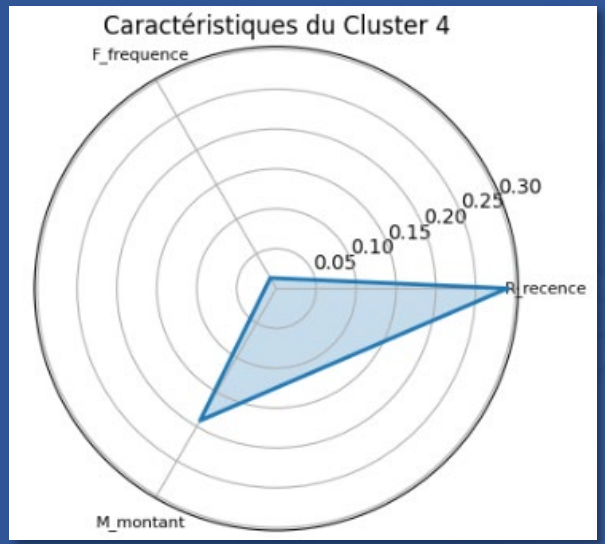
Interprétation des clusters











Centroïds de la CAH :

CAH_label	R_recence	F_frequence	M_montant
1	216.956890	2.120862	377.541955
2	384.408525	1.000000	150.351217
3	127.125812	1.000000	152.909665
4	216.814286	1.085714	7424.719714
5	201.684553	1.017073	1542.449642

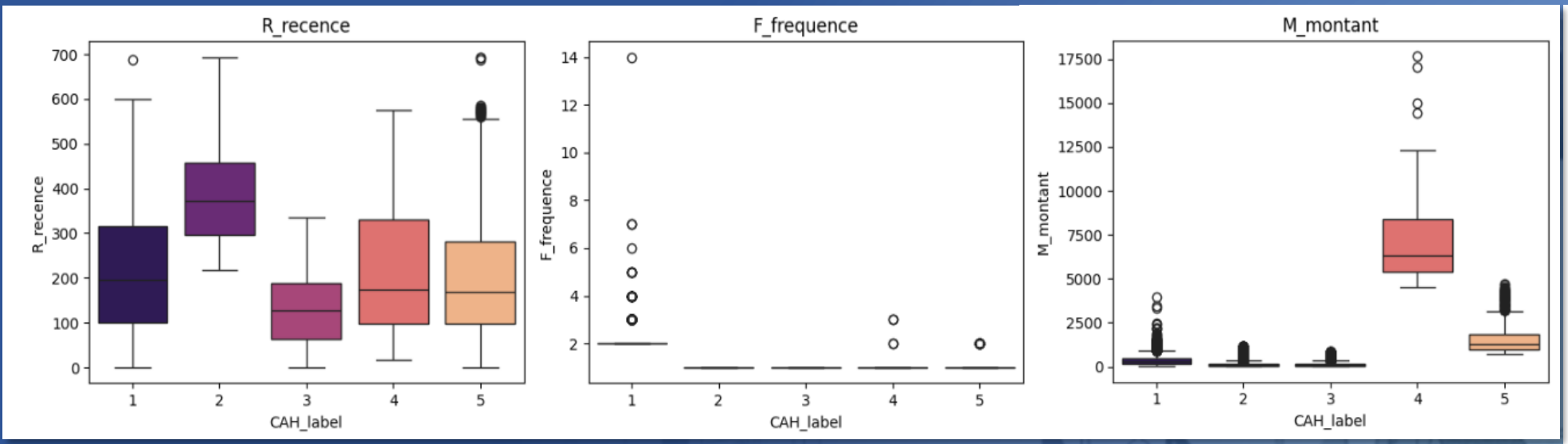
Nombre de client par clusters :

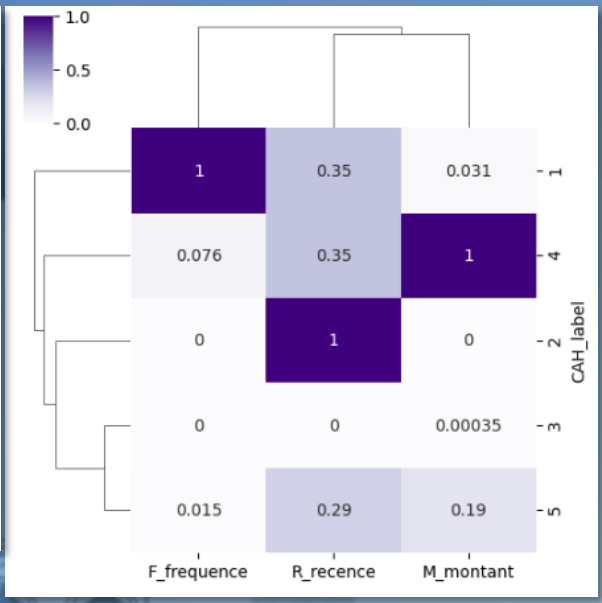
CAH_label	count
3	24314
2	18087
1	1299
5	1230
4	70

olist

CAH avec les données RFM .3/3

Interprétation des clusters





Interprétation métier des clusters :

Le **cluster 1** correspond aux *clients fidèles* ; ces clients ont commandé au moins 2 fois sur le site.

Le **cluster 2** correspond aux *clients perdus*. Ils ont commandé une seule fois pour un montant modéré il y a 377 jours en moyenne.

Le **cluster 3** correspond aux derniers clients ayant commandé ; ils peuvent être assimilés à de *nouveaux clients* car c'était leur première commande.

Le **cluster 4** correspond aux *clients ayant dépensé le plus sur le site* ; dans la majorité des cas ces clients n'ont commandé qu'une seule fois.

Le **cluster 5** correspond aux *clients ayant dépensé une somme importante sur le site* ; dans la majorité des cas ces clients n'ont commandé qu'une seule fois.

olist

DBSCAN avec les données RFM .1/2

Choix du nombre optimal de clusters

Ici, pour trouver le nombre optimal de cluster il faut trouver :

- Epsilon : qui sera la distance maximale entre deux points pour qu'ils soient considérés comme étant dans le même voisinage,
- Et Minimum Points : qui sera le nombre minimum de points pour former un cluster dense.

	min_samples	eps	Silhouette	Davies_Bouldin	Type
10	5	1.0	0.734269	0.847784	DBSCAN
11	5	1.5	0.733813	1.046944	DBSCAN
7	4	1.0	0.733645	1.022659	DBSCAN
2	2	1.5	0.733483	1.075835	DBSCAN
8	4	1.5	0.733469	1.358878	DBSCAN
5	3	1.5	0.733355	1.340815	DBSCAN
1	2	1.0	0.733053	1.131369	DBSCAN
4	3	1.0	0.733000	1.341611	DBSCAN
9	5	0.5	0.492870	1.391978	DBSCAN
6	4	0.5	0.492571	1.458050	DBSCAN
3	3	0.5	0.490833	1.358840	DBSCAN
0	2	0.5	0.483565	1.267555	DBSCAN

On part toujours du principe qu'il faut **maximiser** le score de Silhouette et qu'il faut **minimiser** l'indice de Davies-Bouldin.

On obtient donc :

- **epsilon = 1.0**
- **min_samples = 5**

Nombre de clusters = 6

Nombre de client par clusters :

DBSCAN_label	count
1	88502
0	2470
2	159
-1	64
3	21
4	9

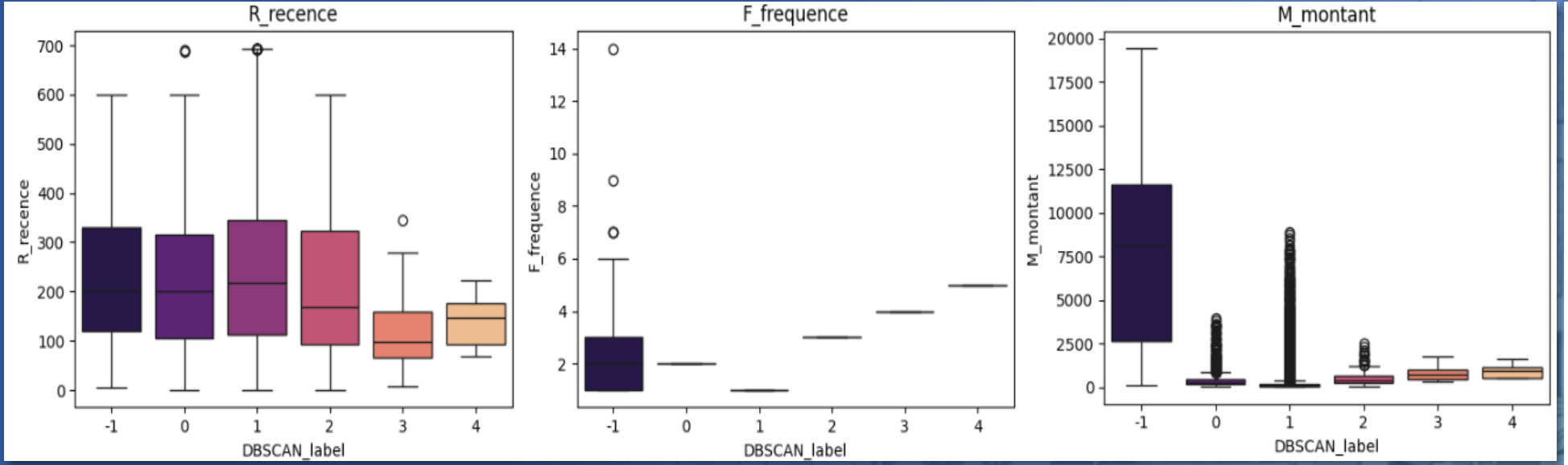
Le premier problème auquel on va se confronter est le nombre de clients par clusters qui est trop déséquilibré.

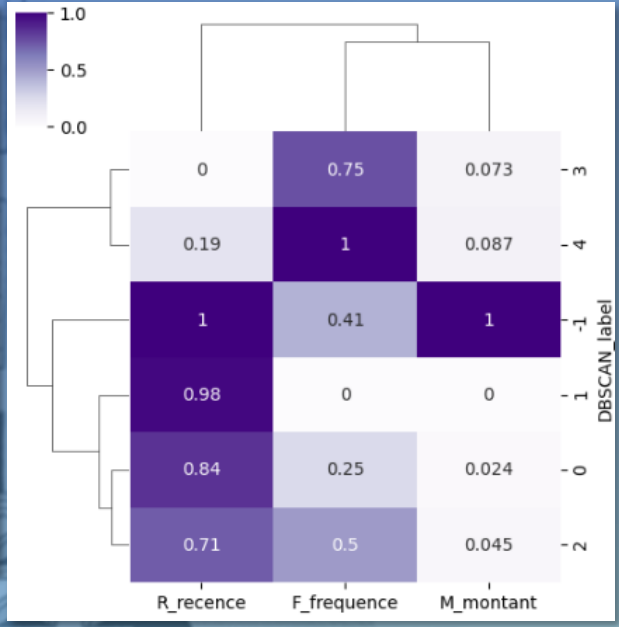
Ceci étant dû à un nombre trop important de clusters.

olist

DBSCAN avec les données RFM .2/2

Interprétation des clusters





Interprétation métier des clusters :

Ici, le DBSAN isole les clients atypiques dans le cluster -1 avec les clients qui ont le plus commandé ou bien les clients qui ont commandé pour de très gros montants (ou les deux).

Mais pour l'interprétation des autres clusters cela devient très compliqué.

Visiblement, le DBSCAN caractérise les différents types de clients par leur fréquence d'achat.

olist

olist

5. Choix du modèle de clustering

Choix du modèle de clustering

Comparaison des scores des modèles

	Type	n_clusters	Silhouette	Davies_Bouldin	min_samples	eps
0	KMeans	5.0	0.525578	0.620122	NaN	NaN
1	CAH	5.0	0.519143	0.632865	NaN	NaN
2	DBSCAN	6.0	0.734269	0.847784	5.0	1.0

Le choix du modèle se portera sur le **KMeans**.

En éliminant le DBSCAN en raison de son interprétation métier moins pratique puis en visant à **maximiser le score de silhouette**, et à **minimiser l'indice de Davies-Bouldin**, le **KMeans** se présente comme une option favorable.

Par ailleurs, le KMeans présente des avantages en termes de performance des ressources, notamment par rapport à une Classification Ascendante Hiérarchique (CAH), pour laquelle nous avons dû échantillonner le jeu de données des clients.

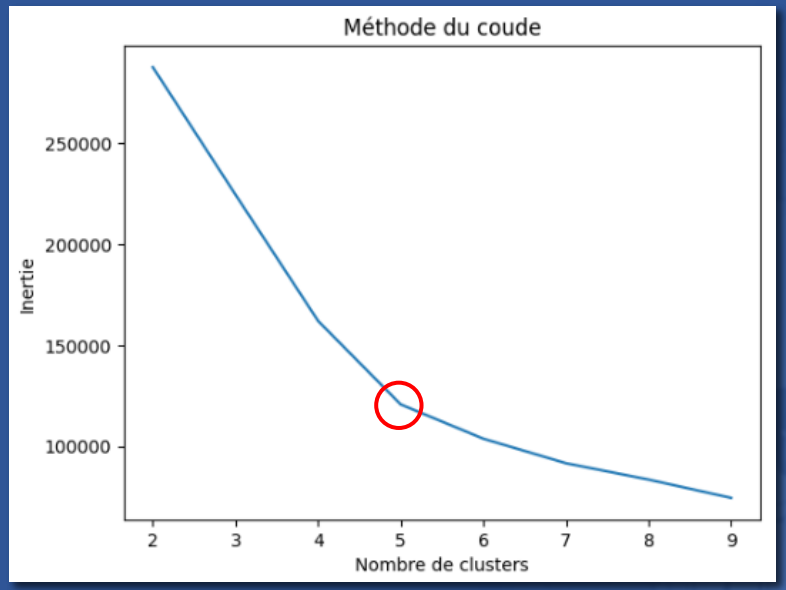
olist

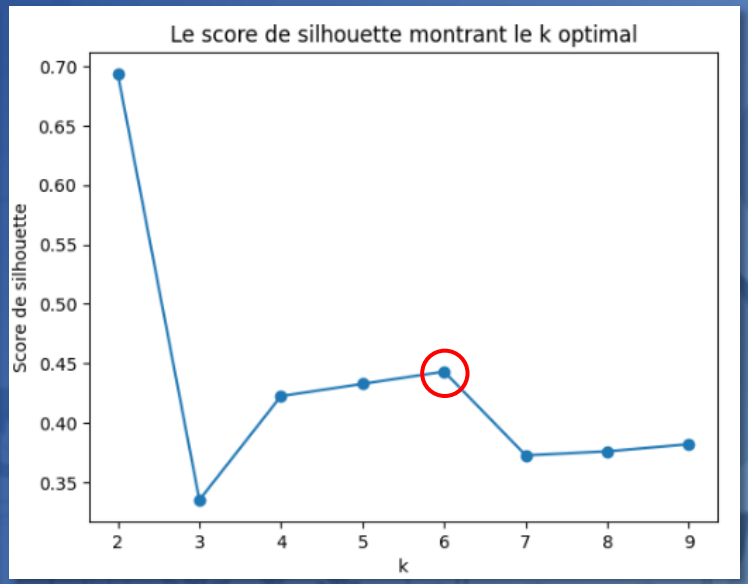
olist

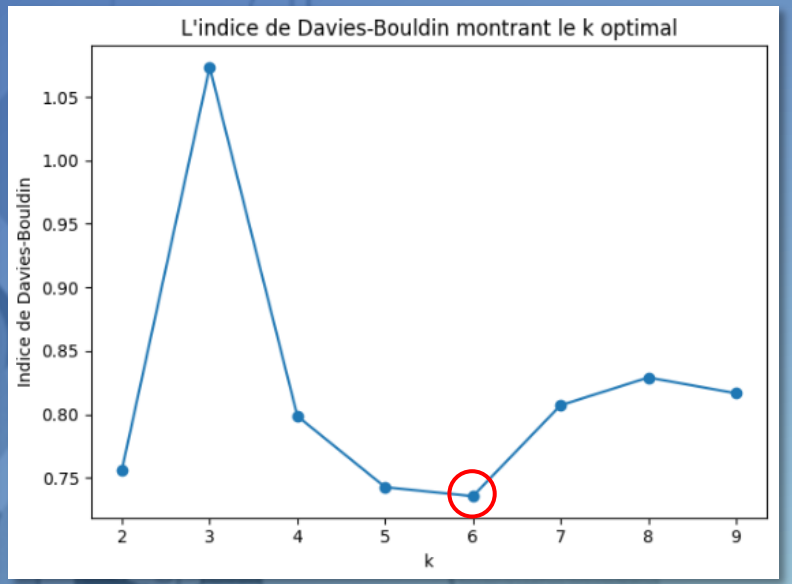
6. Modèle choisi en rajoutant le review score

KMeans avec les données RFM et le Review Score .1/3

Choix du nombre de clusters







	Type	n_clusters	Silhouette	Davies_Bouldin
0	KMeans + ReviewScore	2	0.693680	0.756450
4	KMeans + ReviewScore	6	0.443026	0.735506
3	KMeans + ReviewScore	5	0.432726	0.742658
2	KMeans + ReviewScore	4	0.422399	0.798567

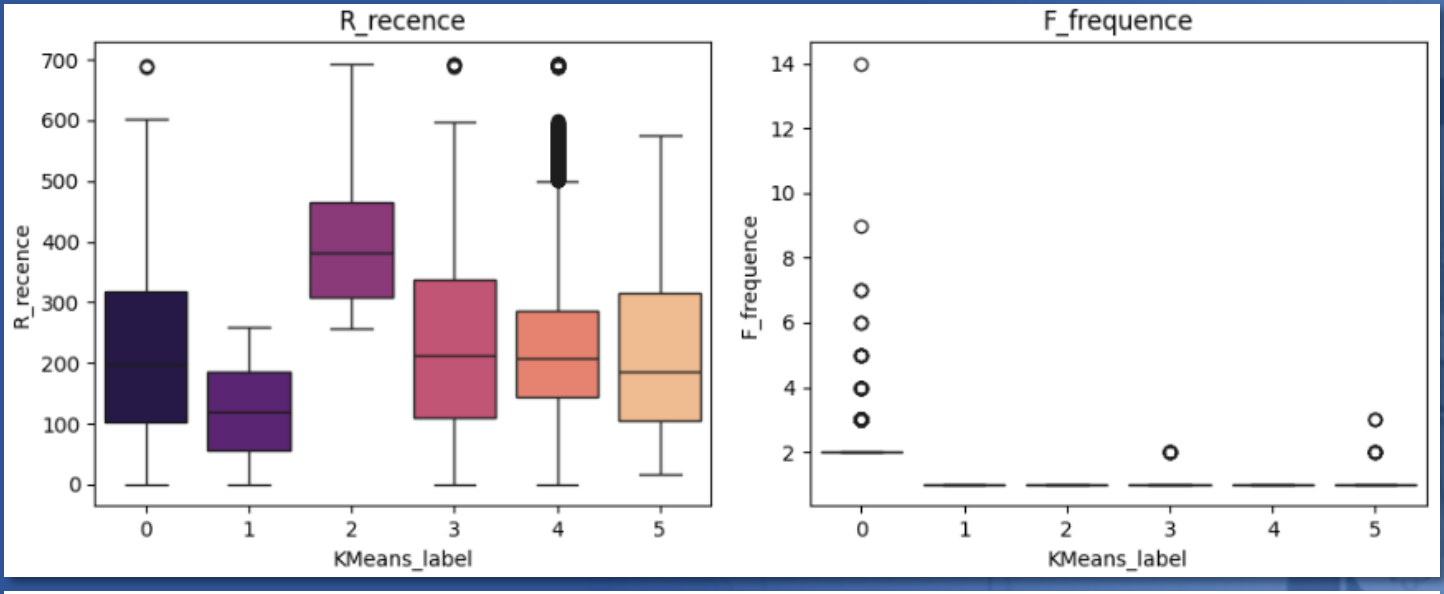
En utilisant la méthode du coude, nous constatons que le nombre idéal de clusters est de 5.

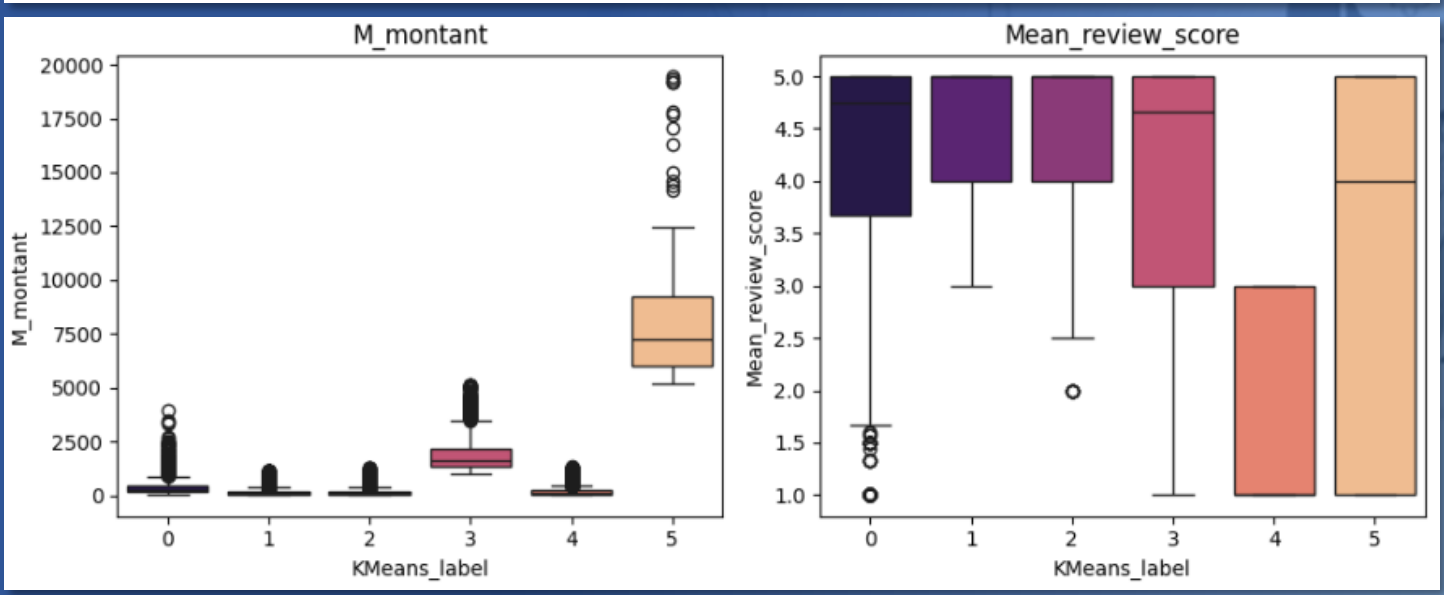
En se basant sur le score de silhouette et sur l'indice de Davies-Bouldin l'option de 6 clusters semble être la meilleur.

olist

KMeans avec les données RFM et le Review Score .2/3

Interprétation des clusters





Interprétation métier des clusters :

Le **cluster 0** correspond aux *clients fidèles* ; caractérisés par des achats fréquents, à noter que ces clients sont majoritairement satisfaits.

Le **cluster 1** correspond aux *nouveaux clients* ; ces clients ont commandé une seule fois pour un montant modéré et ils sont satisfaits.

Le **cluster 2** correspond aux *clients perdus*. Ils ont commandé une seule fois pour un montant modéré il y a 393 jours en moyenne. Ils étaient satisfait de leur expérience.

Le **cluster 3** correspond aux *clients ayant dépensé une somme importante sur le site* ces clients sont majoritairement satisfaits.

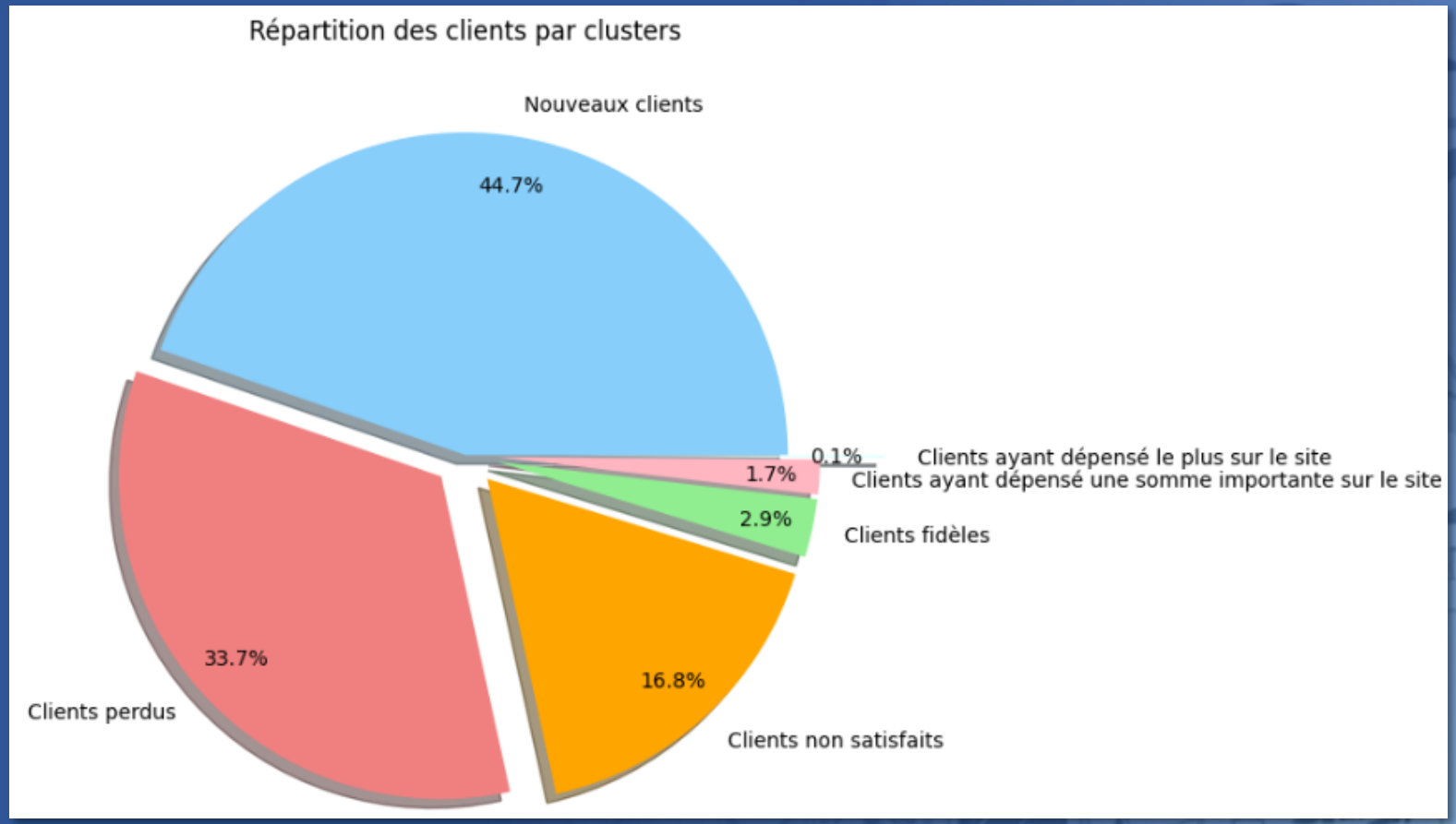
Le **cluster 4** correspond aux *clients non satisfaits*.

Le **cluster 5** correspond aux *clients ayant dépensé une somme importante sur le site* ; dans la majorité des cas ces clients n'ont commandé qu'une seule fois.

olist

KMeans avec les données RFM et le Review Score .3/3

Répartition des clients par clusters



Cette option qui consiste à rajouter le review score nous apporte une précision accrue en termes d'interprétation et nous permet d'obtenir un cluster supplémentaire qui facilite l'identification des clients mécontents.

De plus, cela nous permet de connaître le niveau de satisfaction sur les autres clusters pour une meilleure interprétation métier.

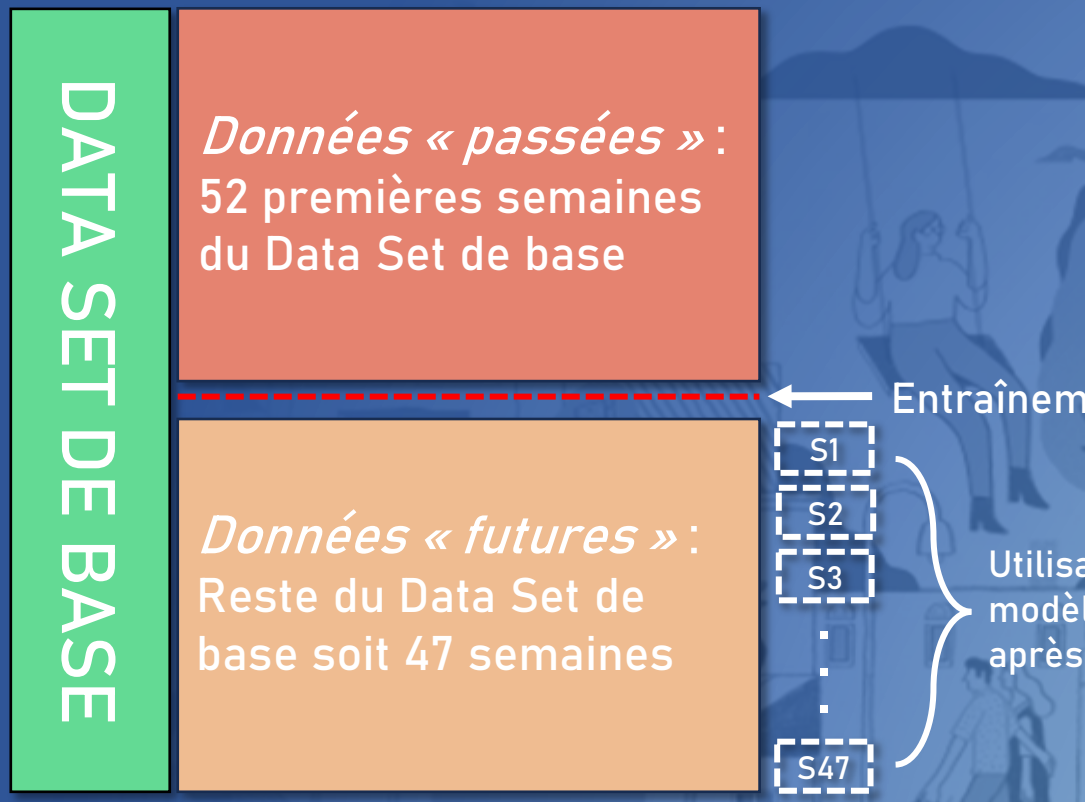
olist

olist

7. Contrat de maintenance

Simulation de données « passées » et « futures »

Séparation du Data Set en 2 et explication de la technique de simulation utilisées



Le but est de séparer notre Data Set en deux, d'entraîner le modèle choisi sur les données qui simuleront « le passé » et de voir comment le modèle se comporte sur les données « futures » sans réentraînement.

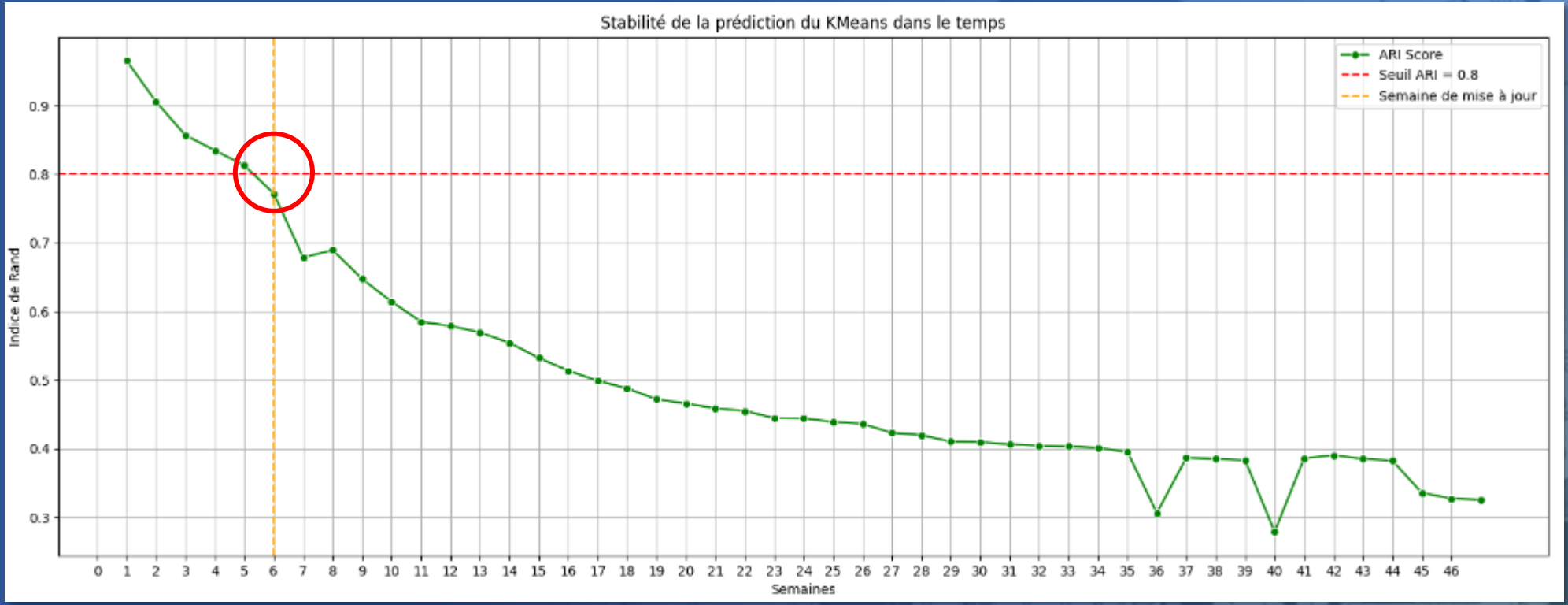
Il faudra le confronter au même modèle que nous réentraîneront toutes les semaines afin de simuler la stabilité de notre modèle de base à l'incorporation de données futures.

Pour comparer les résultats des deux modèles nous utiliseront l'indice de Rand (Adjusted Rand Index (ARI)).

olist

Fréquence nécessaire de mise à jour du modèle

Stabilité des prédictions du KMeans dans le temps



	Date	Semaine	ARI_Score
0	2018-08-26 17:00:40	1	0.965457
1	2018-08-26 17:00:40	2	0.905417
2	2018-08-26 17:00:40	3	0.856606
3	2018-08-26 17:00:40	4	0.834628
4	2018-08-26 17:00:40	5	0.812665
5	2018-08-26 17:00:40	6	0.771691
6	2018-08-26 17:00:40	7	0.678334
7	2018-08-26 17:00:40	8	0.689391
8	2018-08-26 17:00:40	9	0.647253
9	2018-08-26 17:00:40	10	0.614169

Si on prend le seuil de 0.8, la mise à jour du modèle devrait être effectuée à la semaine 6, soit 1 mois et demi après l'avoir entraîné sur les données.

olist