

Segmentez des clients d'un site e- commerce

Sarah Bitan

Introduction

Je suis ravi de vous présenter aujourd'hui le projet passionnant sur lequel nous travaillons en collaboration avec Olist, une entreprise brésilienne innovante dans le domaine de la vente sur les marketplaces en ligne.

Cette mission s'inscrit dans le cadre d'un scénario professionnel où nous agissons en tant que consultants pour accompagner Olist dans la mise en place de leur équipe Data et leur premier cas d'usage en Data Science, axé sur la segmentation client.

Pour mener à bien cette mission, nous avons défini un ensemble d'étapes claires pour vous guider tout au long du processus :



Plan

01.

Implémentation des Requêtes SQL : Suite à notre intégration chez Olist, notre première tâche consistait à répondre à l'urgence en aidant Fernanda, Lead Data Analyst, dans la construction et la maintenance du Dashboard Customer Experience en intégrant des requêtes SQL essentielles.

02.

Analyse Exploratoire des Données : Notre objectif principal est de fournir à l'équipe Marketing d'Olist une segmentation des clients basée sur leur comportement et leurs données personnelles. Pour cela, nous avons entrepris une analyse exploratoire des données fournies par l'entreprise depuis janvier 2017.

03.

Modélisation et Maintenance de la Segmentation : Nous avons exploré différentes approches de modélisation, dont le clustering non supervisé, pour identifier les différents types d'utilisateurs. De plus, nous avons simulé la fréquence nécessaire de mise à jour du modèle de segmentation pour garantir sa pertinence dans le temps.

Implémentation des Requêtes SQL

Dans cette section, nous aborderons notre première tâche chez Olist : l'intégration des requêtes SQL essentielles pour la construction et la maintenance du Dashboard Customer Experience.

1. En excluant les commandes annulées, quelles sont les commandes récentes de moins de 3 mois que les clients ont reçues avec au moins 3 jours de retard ?



```
Requête Historique
1 ---En excluant les commandes annulées, quelles sont les commandes récentes de moins de 3 mois que les clients ont reçues a
2
3
4 SELECT o.order_id, o.order_status, o.order_purchase_timestamp, o.order_estimated_delivery_date, o.order_delivered_customer
5 FROM orders o
6 JOIN customers c ON o.customer_id = c.customer_id
7 WHERE o.order_status = 'delivered' -- Commandes livrées
8 AND o.order_delivered_customer_date >= date(o.order_purchase_timestamp, '-3 months') -- Commandes livrées dans les 3 derni
9 AND DATE(o.order_delivered_customer_date) > DATE(o.order_estimated_delivery_date, '+3 days'); -- Commandes reçues avec au
10

Table Formulaire
Nombre de lignes chargées : 4664
order_id order_sta order_purchase_time order_estimated_del order_delivered_cust customer_id cu
1 203096f03d82e0dffbc41ebc2e2bcfb7 delivered 2017-09-18 14:31:30 2017-09-28 00:00:00 2017-10-09 22:23:46 d2b091571da224a1b36412c18bc3bbfe sa
2 fbf9ac61453ac646ce8ad9783d7d0af6 delivered 2018-02-20 23:46:53 2018-03-12 00:00:00 2018-03-21 22:03:54 3a874b4d4c4b6543206ff5d89287f0c3 rie
3 6ea2f835b4556291ffdc53fa0b3b95e8 delivered 2017-11-24 21:27:48 2017-12-21 00:00:00 2017-12-28 18:59:23 c7340080e394356141681bd4c9b8fe31 pri
4 a685d016c8a26f71a0bb67821070e398 delivered 2017-03-13 18:14:36 2017-03-30 00:00:00 2017-04-06 13:37:16 911e4c37f5cafe1604fe6767034bf1ae ca
5 6a0a8bfbbe700284feb0845d95e0867f delivered 2017-11-22 11:32:22 2017-12-11 00:00:00 2017-12-28 19:43:00 68451b39b1314302c08c65a29f1140fc rie
6 9d531c565e28c3e0d756192f84d8731f delivered 2017-11-28 21:00:44 2017-12-22 00:00:00 2018-01-23 21:38:52 d4faa220408c20e53595d2950f361f3b m
7 8fc207e94fa91a7649c5a5dab690272a delivered 2017-11-26 17:49:46 2017-12-19 00:00:00 2018-01-20 13:42:22 c69f8b33e62ecb30ff78ae46d7fb9241 sa
8 33a3edb84b9df4cb49546859b990ac6d delivered 2018-02-21 17:15:49 2018-03-16 00:00:00 2018-03-22 00:03:53 35ec6c1ca9e5844c5ca94214cce16dca ju
```

Qui sont les vendeurs ayant généré un chiffre d'affaires de plus de 100 000 Real sur des commandes livrées via Olist ?

Requête Historique

```
1 ---Qui sont les vendeurs ayant généré un chiffre d'affaires de plus de 100 000 Real sur des commandes livrées via Olist ?
2
3 SELECT seller_id
4 FROM order_items
5 WHERE order_id IN (
6   SELECT order_id
7   FROM orders
8   WHERE order_status = 'delivered' -- Commandes livrées
9 )
10 GROUP BY seller_id
11 HAVING SUM(price + freight_value) > 100000; -- Chiffre d'affaires total supérieur à 100 000 Real
```

Table Formulaire

Nombre de lignes chargées : 19

	seller_id
1	1025f0e2d44d7041d6cf58b6550e0bfa
2	1f50f920176fa81dab994f9023523100
3	3d871de0142ce09b7081e2b9d1733cb1
4	46dc3b2cc0980fb8ec44634e21d2718e
5	4869f7a5dfa277a7dca6462dcf3b52b2
6	4a3ca9315b744ce9f8e9374361493884
7	53243585a1d6dc2643021fd1853d8905
8	5dceca129747e92ff8ef7a997dc4f8ca
9	620c87c171fb2a6dd6e8bb4dec959fc6
10	6560211a19b47992c3666cc44a7e94c0
11	7a67c85e85bb2ce8582c35f2203ad736

Qui sont les nouveaux vendeurs (moins de 3 mois d'ancienneté) qui sont déjà très engagés avec la plateforme (ayant déjà vendu plus de 30 produits) ?

```
SELECT s.seller_id,s.seller_city,s.seller_state,s.seller_zip_code_prefix,count(o.order_id) AS engagement FROM sellers AS s
LEFT JOIN order_items AS oi ON s.seller_id=oi.seller_id
LEFT JOIN orders AS o ON oi.order_id=o.order_id
WHERE s.seller_id IN (SELECT s.seller_id FROM sellers AS s
LEFT JOIN order_items AS oi ON s.seller_id=oi.seller_id
LEFT JOIN orders AS o ON oi.order_id=o.order_id WHERE o.order_delivered_customer_date>datetime('2018-10-17 13:22:46','-3 months')
AND s.seller_id NOT IN (SELECT s.seller_id FROM sellers AS s
LEFT JOIN order_items AS oi ON s.seller_id=oi.seller_id
LEFT JOIN orders AS o ON oi.order_id=o.order_id WHERE o.order_delivered_customer_date<datetime('2018-10-17 13:22:46','-3 months')
AND o.order_status='delivered'
GROUP BY s.seller_id HAVING count(o.order_id)>30
```

Quels sont les 5 codes postaux, enregistrant plus de 30 commandes, avec le pire review score moyen sur les 12 derniers mois ?

Requête Historique

```
8 LEFT JOIN order_reviews orv ON o.order_id = orv.order_id
9 WHERE o.order_purchase_timestamp >= (
10    SELECT DATE(MAX(order_purchase_timestamp), '-12 months') -- Date il y a 12 mois par rapport à la
11    FROM orders
12 )
13 GROUP BY c.customer_zip_code_prefix
14 HAVING total_orders > 30
15 ORDER BY avg_review_score ASC
16 LIMIT 5;
17
```

Table Formulaire

Nombre de lignes chargées : 5

	customer_zip_code_prefix	avg_review_score	total_orders
1	22753	2.80851063829787	49
2	22770	3.13513513513514	37
3	22793	3.23333333333333	90
4	21321	3.277777777777778	37
5	22780	3.35135135135135	38

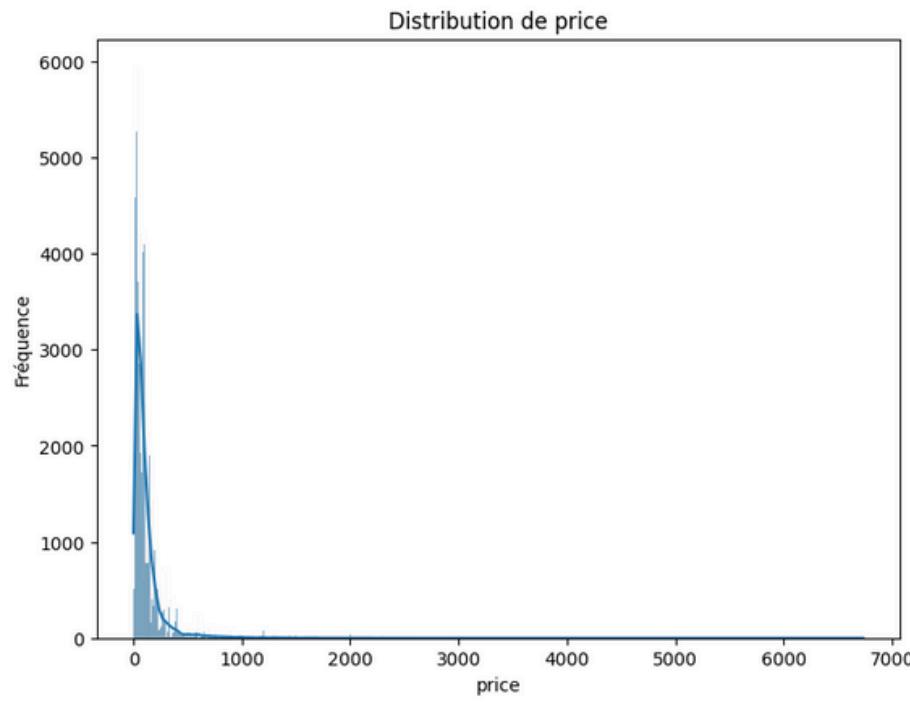
Analyse Exploratoire des Données

Présentation du jeu de données

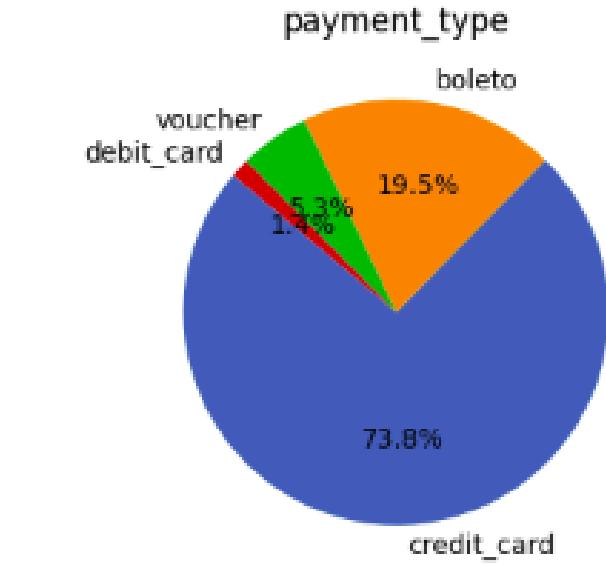
customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state	index_order	order_id	order_status
06b8999e2fba1a1fb88172c00ba8bc7	861eff4711a542e4b93843c6dd7febb0	14409	franca	SP	88492	00e7ee1b050b8499577073aeb2a297a1	delivered
8912fc0c3bbf1e2fbf35819e21706718	9eae34bbd3a474ec5d07949ca7de67c0	68030	santarem	PA	60047	c1d2b34febe9cd269e378117d6681172	delivered
8912fc0c3bbf1e2fbf35819e21706718	9eae34bbd3a474ec5d07949ca7de67c0	68030	santarem	PA	60047	c1d2b34febe9cd269e378117d6681172	delivered
f0ac8e5a239118859b1734e1087ccb1f	3c799d181c34d51f6d44bbc563024db	92480	nova santa rita	RS	42931	b1a5d5365d330d10485e0203d54ab9e8	delivered
6bc8d08963a135220ed6c6d098831f84	23397e992b09769faf5e66f9e171a241	25931	mage	RJ	32077	2e604b3614664aa66867856dba7e61b7	delivered
...
0fbdb856ba1d4961786fb54bd448eb7fe	96328ac15f58fbb232fe14b182103382	89675	vargem bonita	SC	59073	937592924b66482b823ee7ecd185d0ff	delivered
98a1b4f80dd0ccb7d1ae5a75ba5c904e	bf5ed75fca98e8f79c49e9a5bf7690e1	13480	limeira	SP	42587	28915ae2a90c218f0c2b7f8e0fb280e0	delivered
aa0533eb31ed3be79086f11bb2bec430	a490d5875edefe9bb8f5101ec2f6b56f	13870	sao joao da boa vista	SP	4272	e22a3e8048469ea68906f666d446c25c	delivered
d11524bb77c28efad04e4467eac8a660	6968d41eb700f1ea39424e04b854bf7e	30130	belo horizonte	MG	98342	1ce0acf125f1bcd636276dd213363196	delivered
d11524bb77c28efad04e4467eac8a660	6968d41eb700f1ea39424e04b854bf7e	30130	belo horizonte	MG	98342	1ce0acf125f1bcd636276dd213363196	delivered

115609 rows × 47 columns

Analyse des variables quantitatives

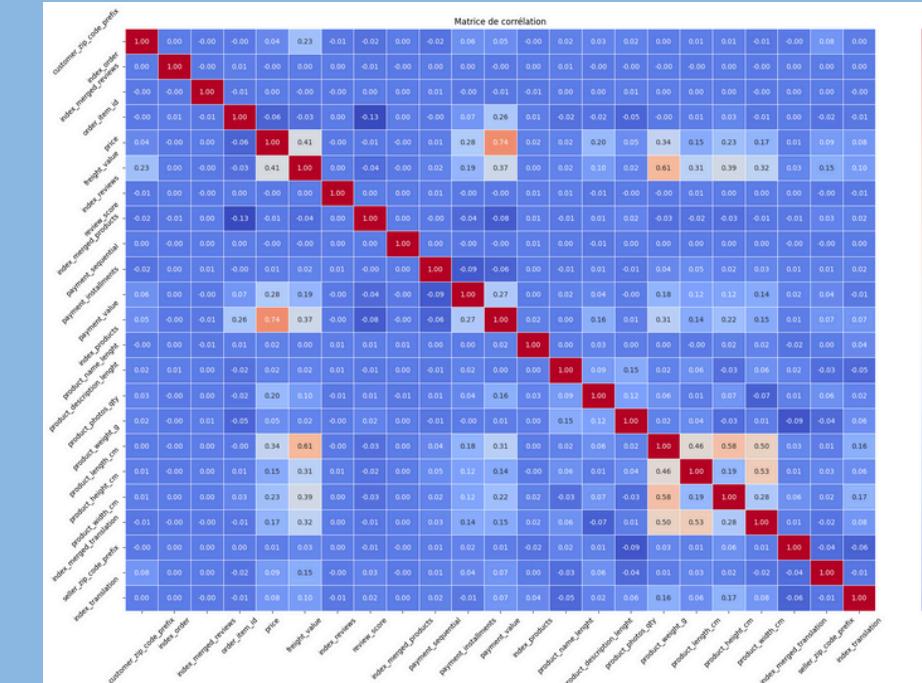


Analyse des variables qualitatives



Analyse univariée et bivariée

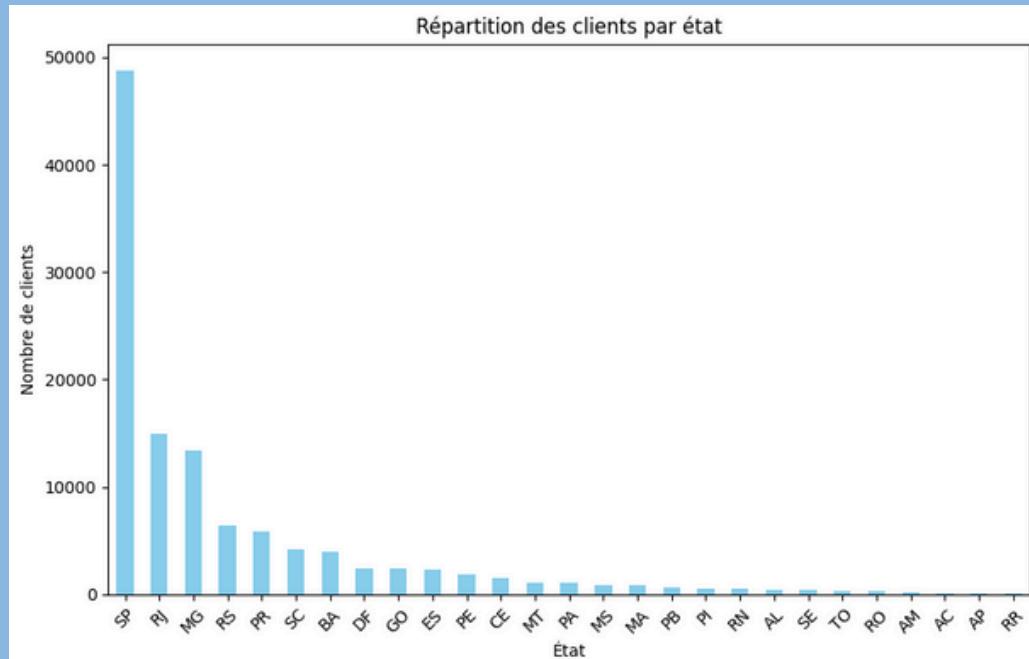
Matrice de corrélation



Répartition des achats

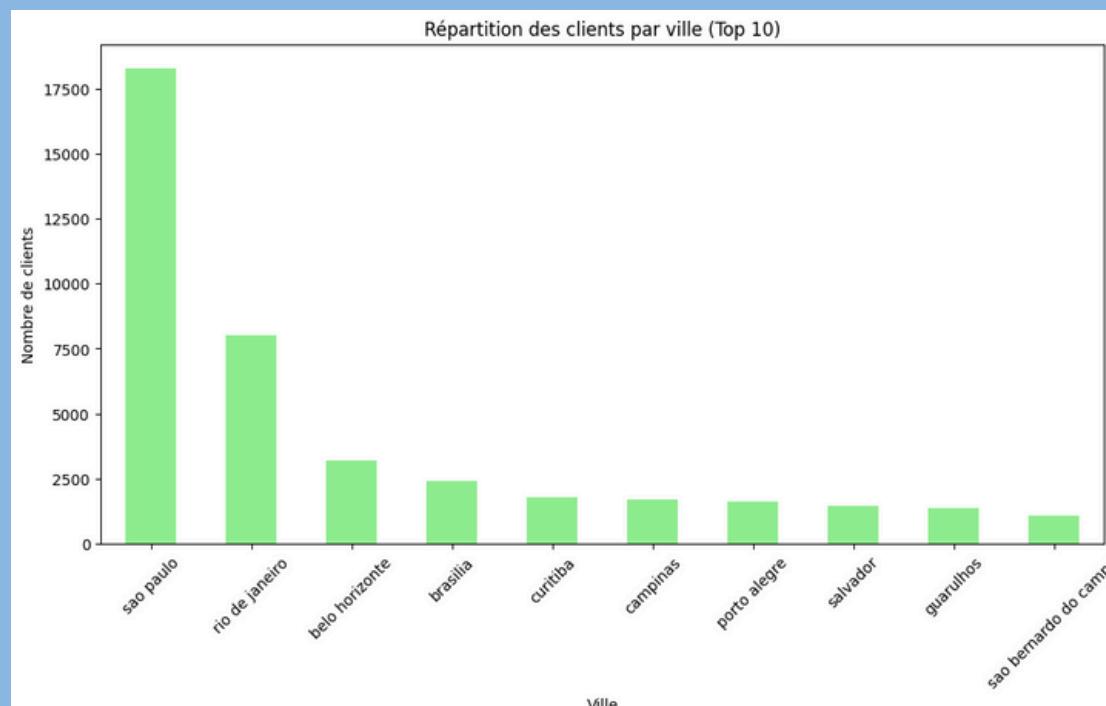


Répartition par état



Répartition des clients

Répartition par ville



Création variables RFM

```
# Calculer le nombre de commandes par client
commandes_par_clients = merged_df.groupby('customer_unique_id')['order_id'].nunique().reset_index()
commandes_par_clients.columns = ['customer_unique_id', 'commandes_par_clients']

# Fusionner avec le DataFrame merged_df
merged_df = pd.merge(merged_df, commandes_par_clients, on='customer_unique_id', how='left')

# Définir la colonne du montant de la commande
montant_commande = merged_df['price']

# Calculer le montant cumulé pour chaque client
merged_df['montant_cumule'] = merged_df.groupby('customer_unique_id')['price'].transform(pd.Series.cumsum)

# Trier le dataframe par date de commande (du plus récent au plus ancien)
df = merged_df.sort_values(by='order_purchase_timestamp', ascending=False)
# Afficher la date de la première ligne (commande la plus récente)
date_commande_plus_recente = df['order_purchase_timestamp'].iloc[0]
# Afficher la date
print(date_commande_plus_recente)

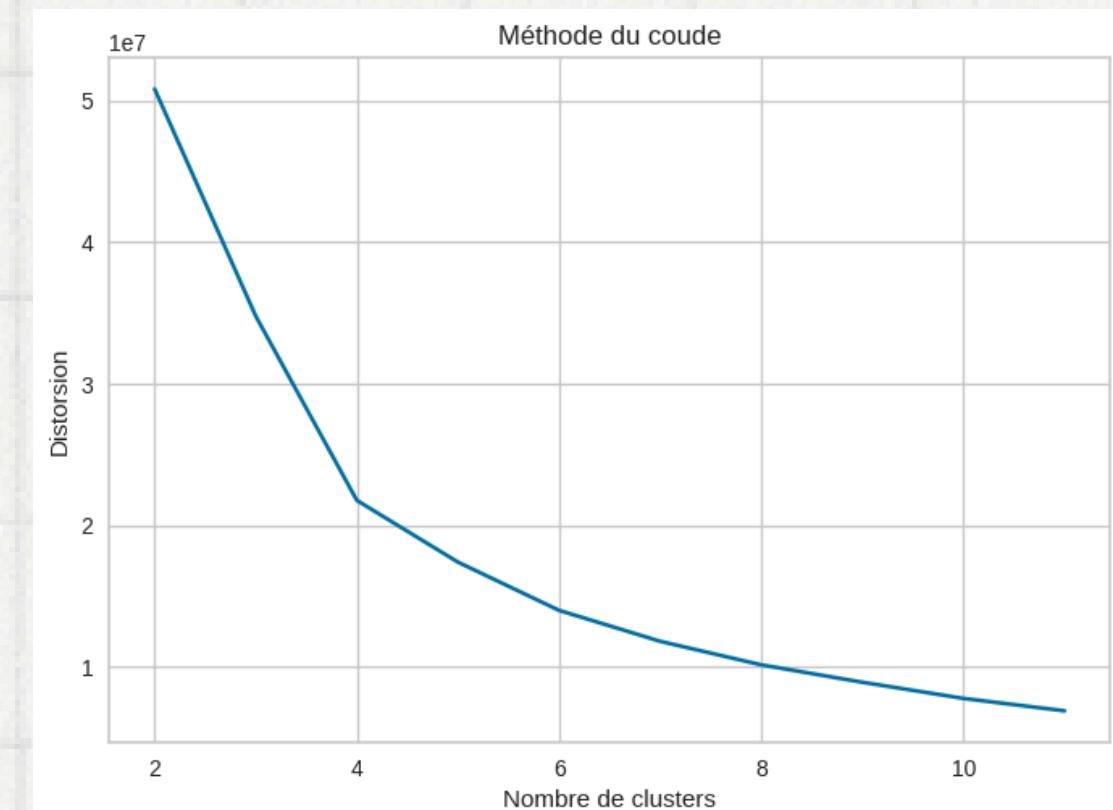
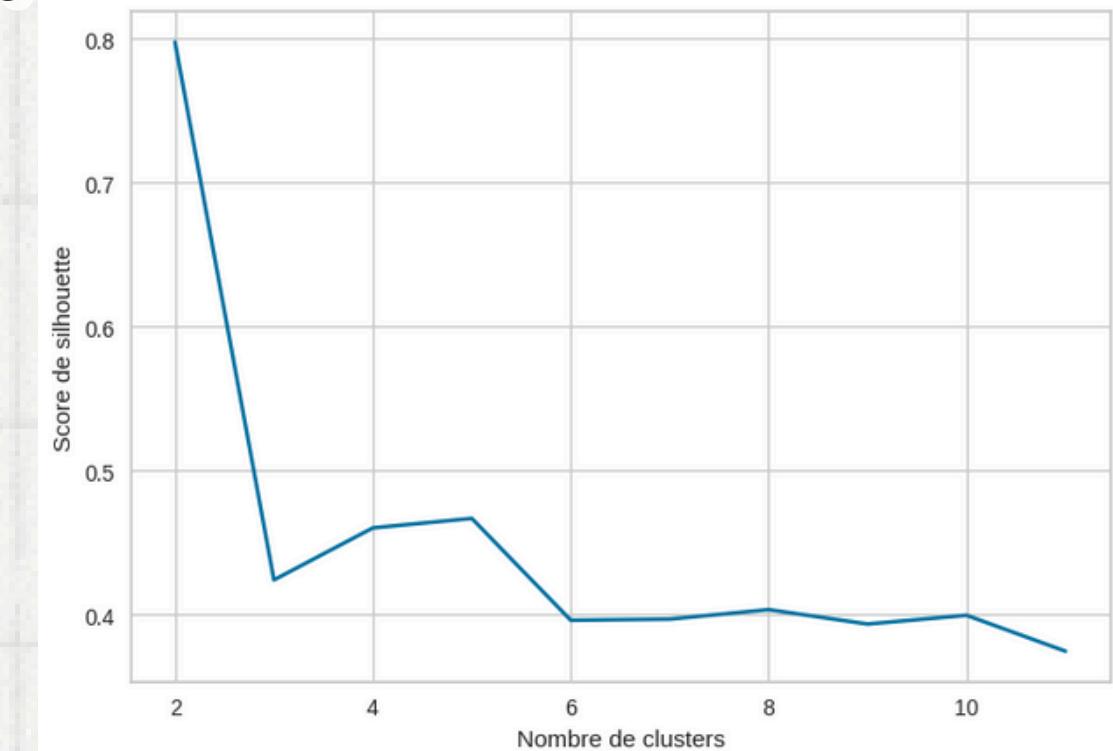
2018-09-03 09:06:57

#Durée depuis la dernière commande pour chaque client
# Convertir la colonne 'date_commande' en type datetime
date_commande = pd.to_datetime(merged_df['order_purchase_timestamp'])
# Définir la date d'aujourd'hui
date_today = pd.to_datetime('2018-09-03')
# Calculer la différence en jours entre la date d'aujourd'hui et la date de la dernière commande
merged_df['duree_depuis_derniere_commande'] = (date_today - date_commande).dt.days
merged_df
```

Modélisation et Maintenance de la Segmentation :

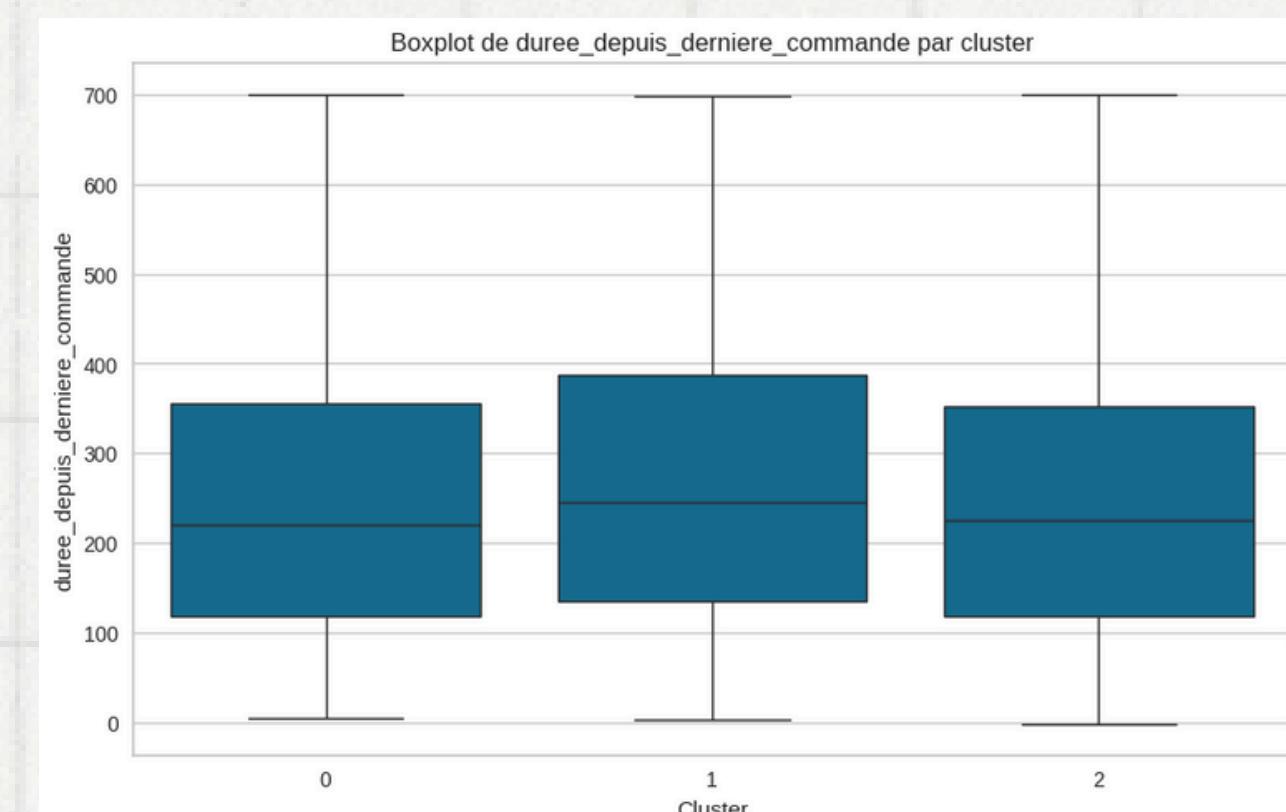
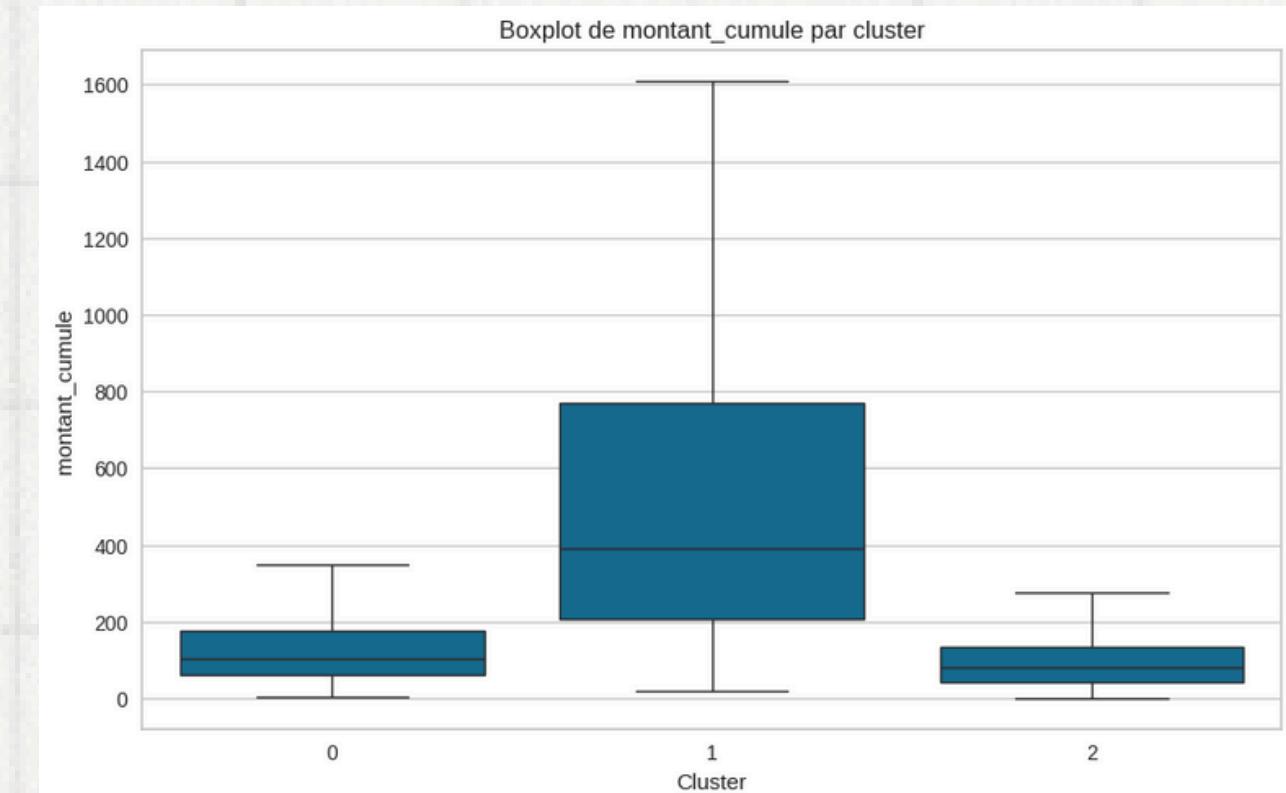
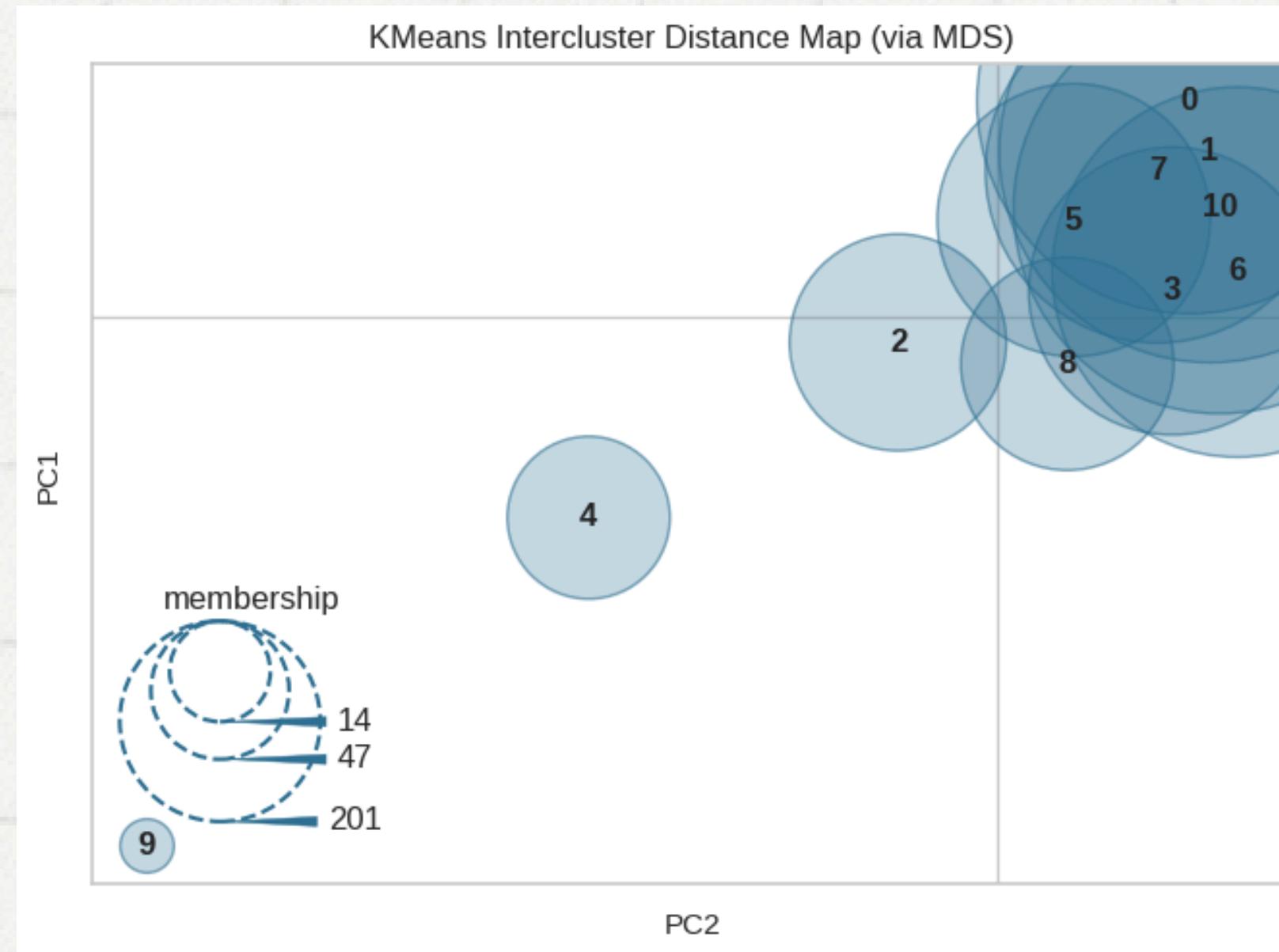
Nous allons tout d'abord réaliser un Kmeans sur les variables RFM.

```
from sklearn.cluster import KMeans  
  
# Définir le nombre de clusters  
n_clusters = 3  
  
# Créer l'instance de KMeans  
kmeans = KMeans(n_clusters=n_clusters)  
  
# Appliquer KMeans aux données transformées  
labels = kmeans.fit_predict(transformed_data)  
  
# Ajouter une nouvelle colonne pour les labels des clusters  
df['cluster'] = labels  
df
```



La carte montre que les données se répartissent en quatre clusters principaux.

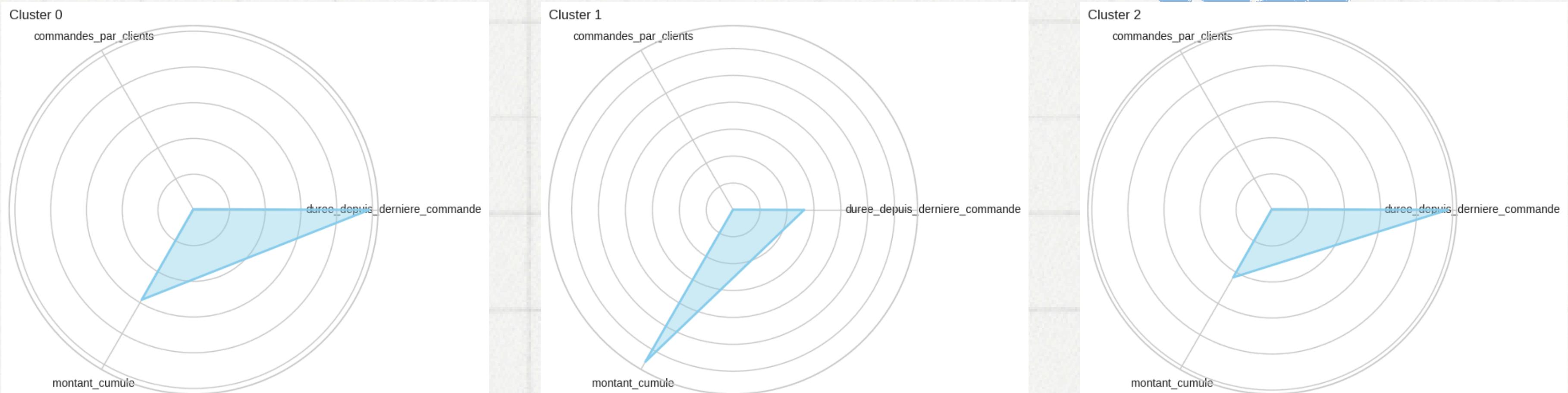
La carte montre également la distance entre les clusters. La distance entre deux clusters est représentée par la longueur du segment qui les relie.



Diagrammes RADAR

Explication du diagramme radar

Le diagramme radar, également appelé diagramme en toile d'araignée ou diagramme de Kiviat, est un type de graphique qui permet de visualiser plusieurs variables quantitatives sur un même plan. Les axes du graphique rayonnent d'un point central et les valeurs des variables sont représentées par des points sur les axes. Les points sont ensuite reliés par des lignes, ce qui permet de visualiser la forme et la position relative des données.



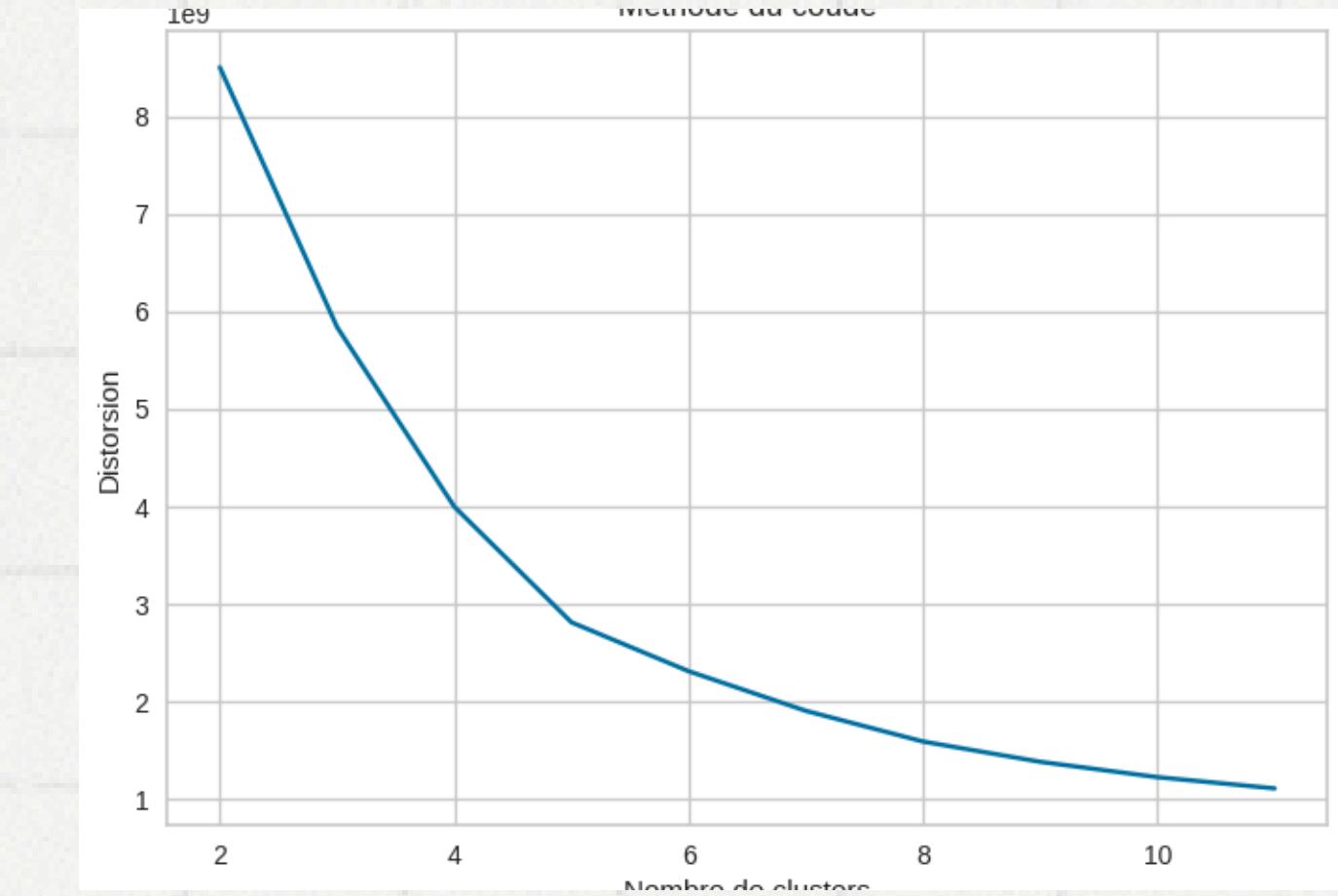
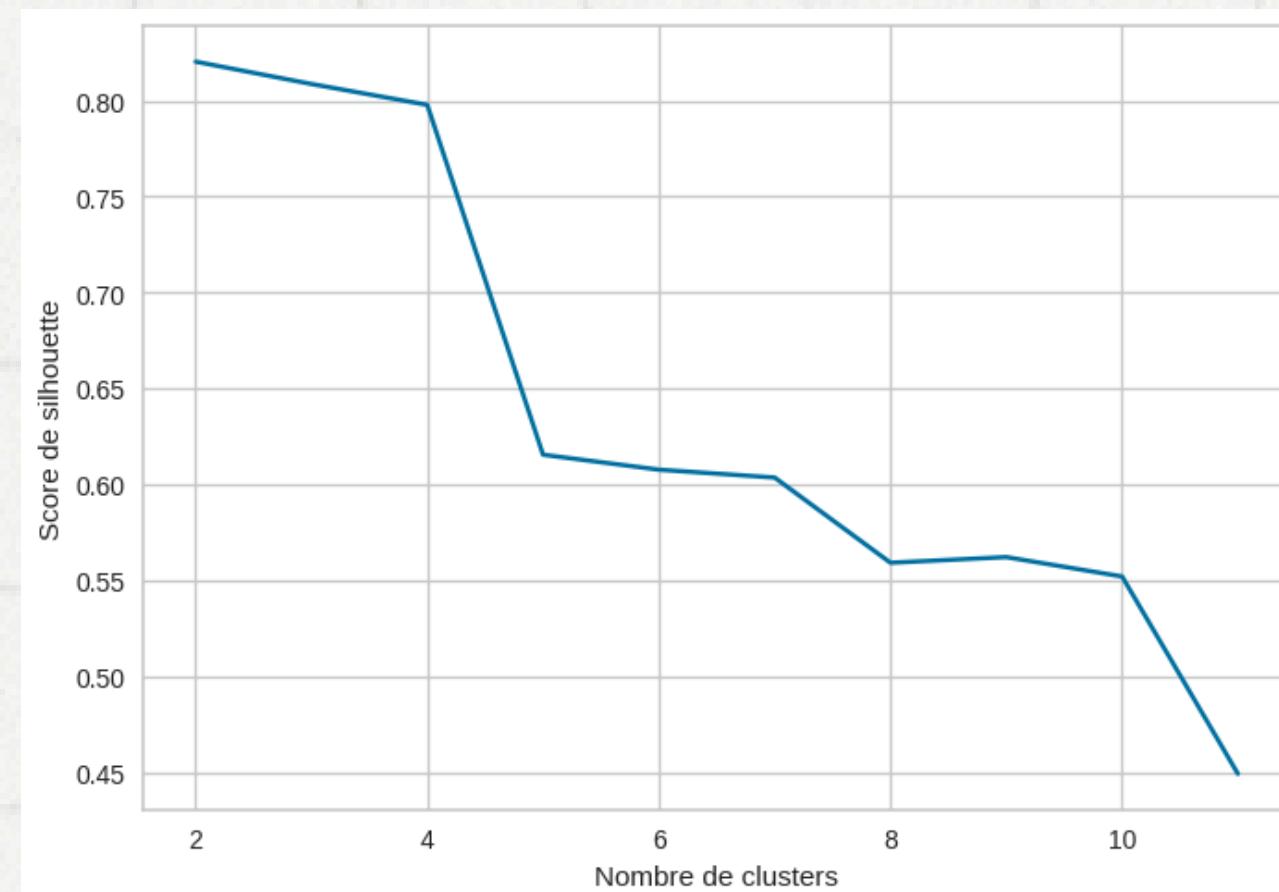
Groupe1: Clients moyens avec durée depuis la dernière commande élevée et un faible montant cumulé

Groupe2: Clients qui ont l'air d'être fidèles avec la durée depuis la dernière commande la plus faible et le montant cuumulé le plus élevé.

Groupe3: Clients avec la durée depuis dernière commande la plus élevée et le montant cummulé le plus faible

On refait le Kmeans en sélectionnant d'autres variables

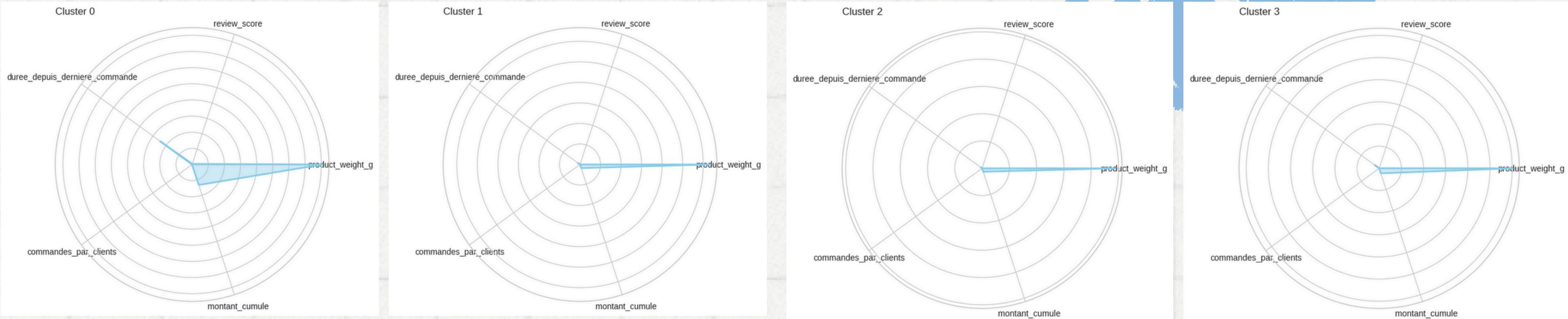
```
df_new = merged_df[['product_weight_g', 'review_score', 'duree_depuis_derniere_commande', 'commandes_par_clients',  
'montant_cumule']]
```



```
[ ] #On réalise notre Kmeans avec le nombre optimal de 4 clusters  
kmeansnew = KMeans(n_clusters=4)  
labels = kmeansnew.fit_predict(df_new)  
  
[ ] #Vérification du nombre d'individus par cluster  
cluster_counts = pd.Series(labels).value_counts().sort_index()  
print(cluster_counts)  
  
0    97516  
1    5351  
2   11591  
3    1151  
Name: count, dtype: int64
```

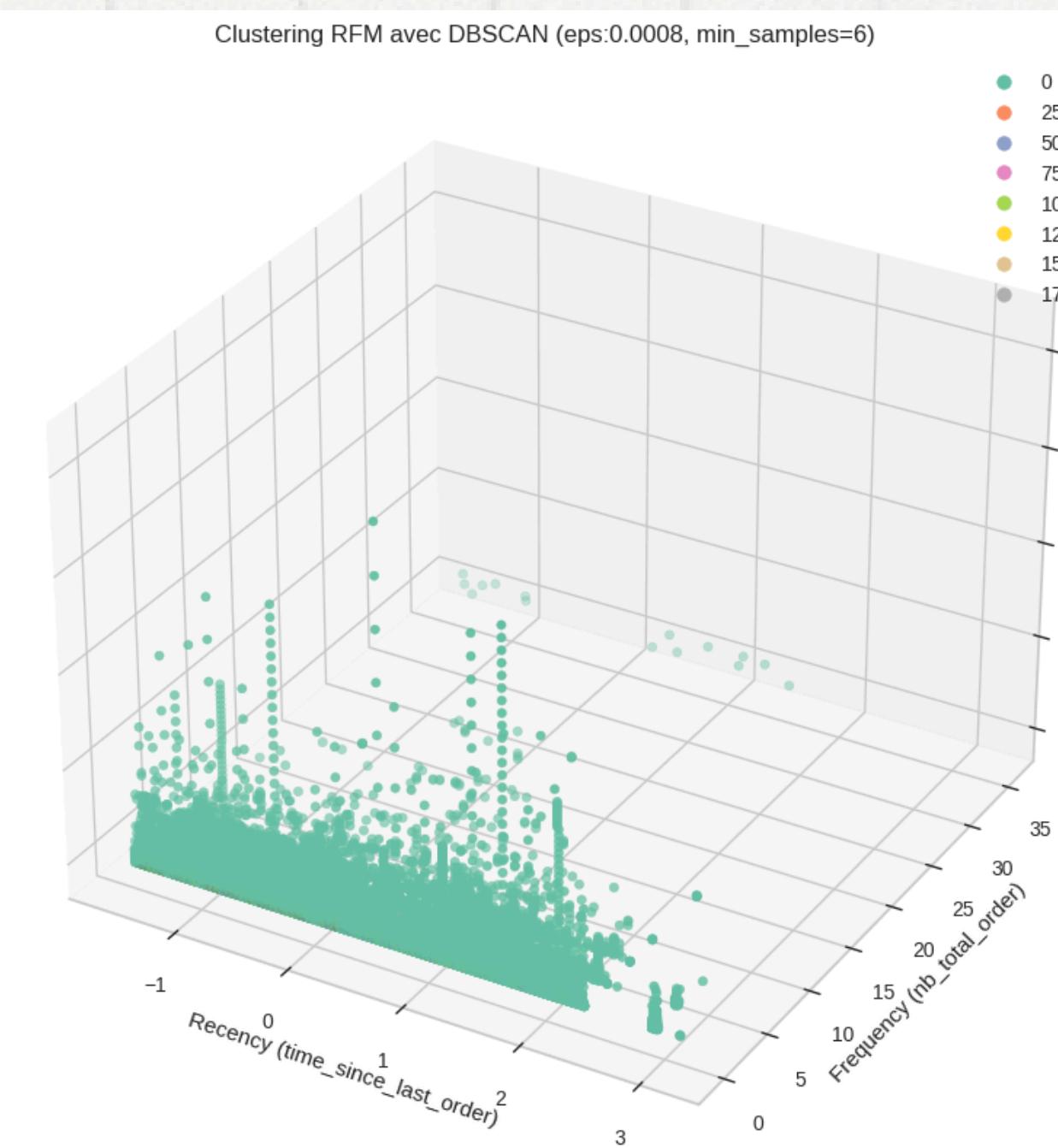
Diagrammes RADAR

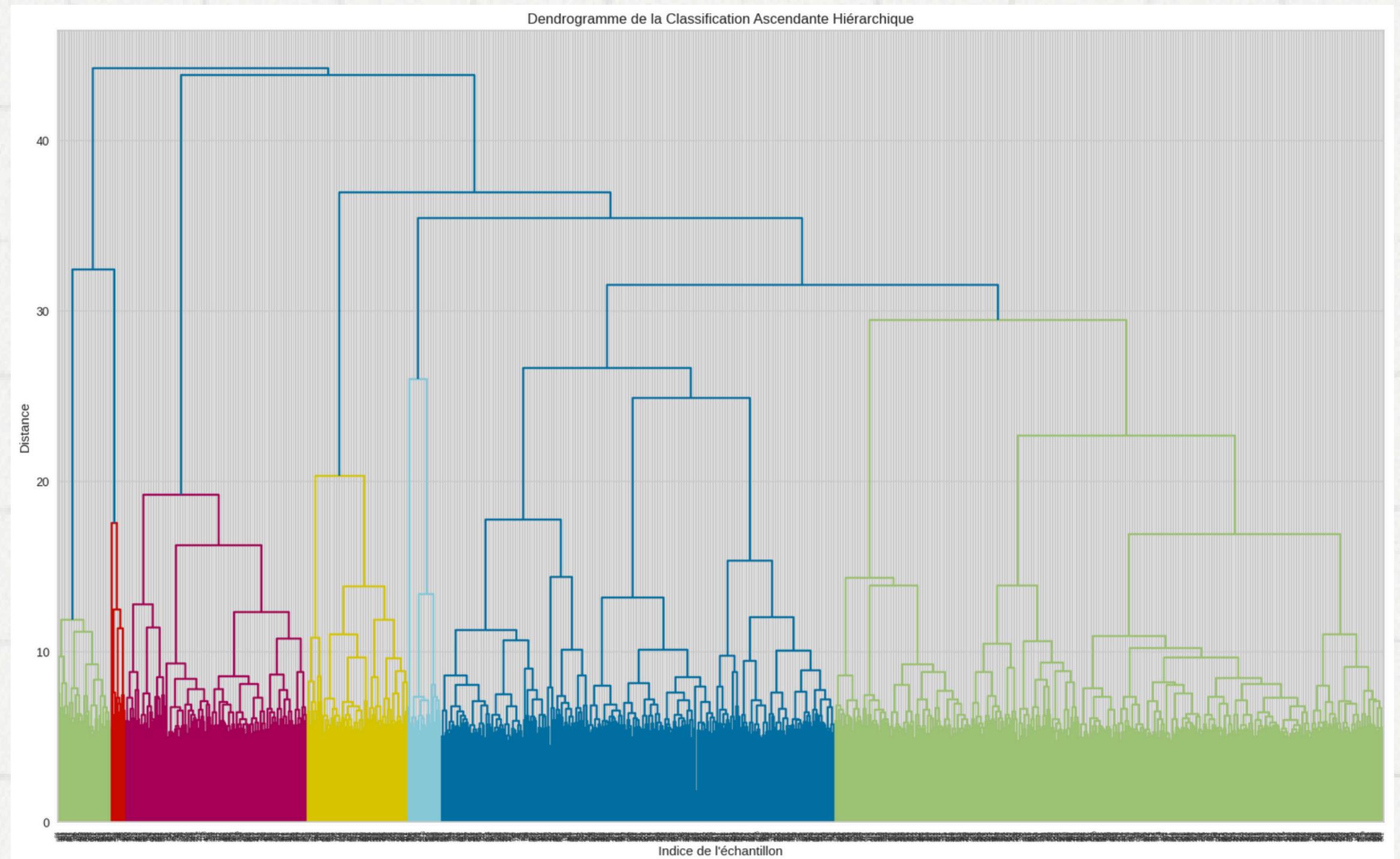
On refait les diagrammes RADAR des nouveaux clusters obtenus avec le k-means



DBSCAN, qui signifie "Density-Based Spatial Clustering of Applications with Noise", est un algorithme de clustering largement utilisé en apprentissage non supervisé pour identifier des clusters dans un ensemble de données. Contrairement à certains autres algorithmes de clustering, comme K-means, DBSCAN ne nécessite pas de spécifier le nombre de clusters à l'avance, ce qui le rend particulièrement utile dans des situations où le nombre de clusters n'est pas connu à l'avance ou peut varier.

- Monetary (Montant) : Représente le montant total des commandes passées par le client.
- Frequency (Fréquence) : Représente le nombre total de commandes passées par le client.
- Recency (Récence) : Représente le temps écoulé depuis la dernière commande du client.

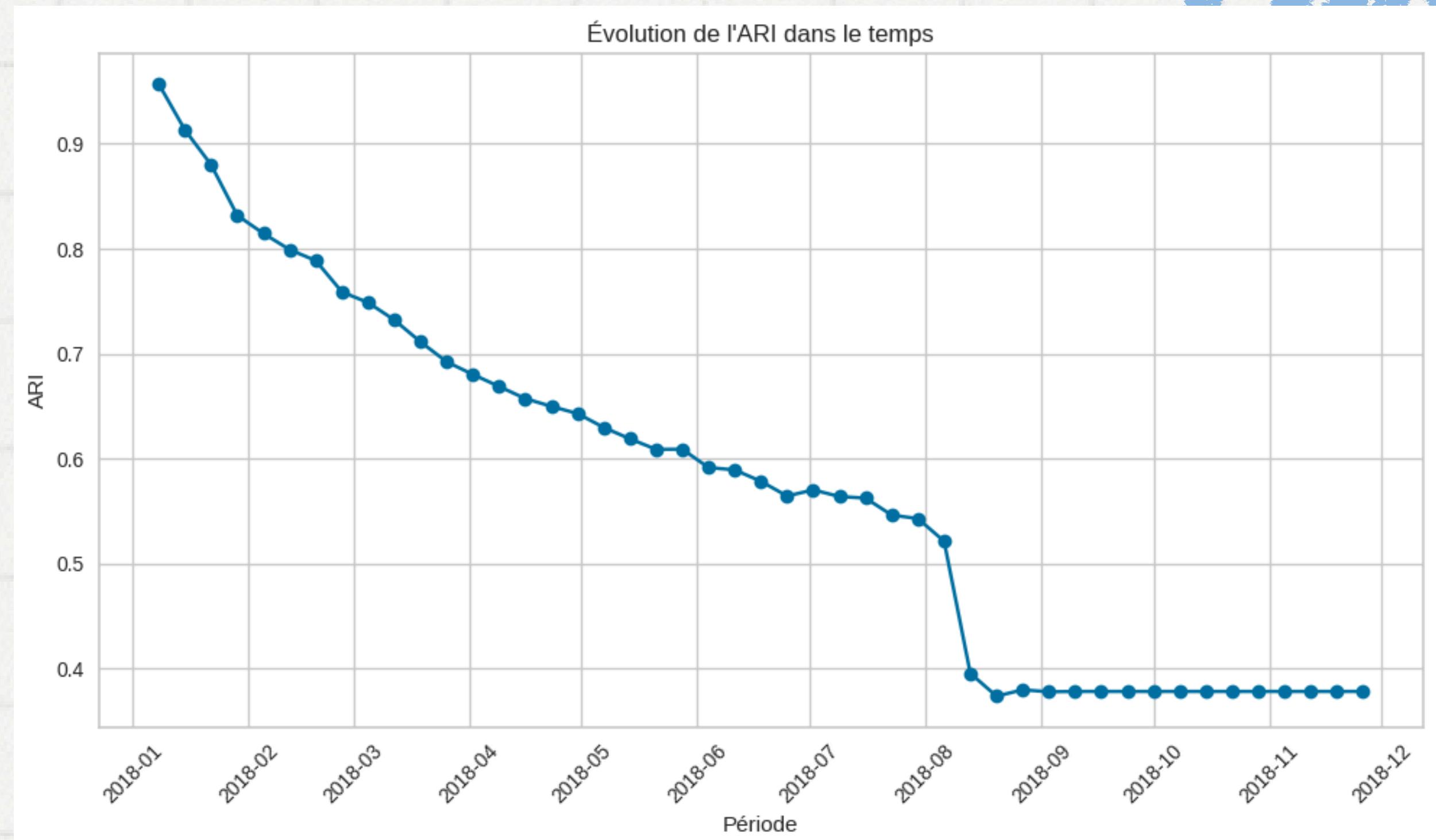




La CAH utilise des mesures de similarité pour regrouper les observations en clusters hiérarchiques. Le dendrogramme est construit en montrant les étapes de regroupement à chaque niveau de la hiérarchie. À chaque étape, les clusters les plus similaires sont regroupés, et le dendrogramme montre visuellement comment ces regroupements sont effectués.

Stabilité des clusters au cours du temps

L'ARI, ou Adjusted Rand Index est une mesure de similarité entre deux ensembles de données regroupés (clusters) qui tient compte des regroupements aléatoires. Il est souvent utilisé pour évaluer la qualité d'un algorithme de clustering en comparant ses résultats aux véritables regroupements (s'il y en a) des données.



Conclusion

Clustering

Notre analyse approfondie des données fournies par Olist nous a permis de réaliser une segmentation des clients basée sur leur comportement d'achat et leurs caractéristiques personnelles. En utilisant l'algorithme de clustering K-means, nous avons identifié quatre clusters distincts, chacun représentant un profil spécifique de client.

Segmentation

Nous avons observé que ces clusters présentent des différences significatives en termes de récence, de fréquence et de valeur monétaire des achats, ce qui permet à l'équipe Marketing d'Olist de mieux comprendre les différents types de clients et d'adapter leurs stratégies de communication en conséquence.

Simulation

En utilisant des méthodes de simulation, nous avons déterminé qu'une mise à jour du modèle de segmentation tous les six mois serait suffisante pour maintenir sa pertinence dans le temps.

Autres approches

Nous avons également exploré d'autres approches de modélisation telles que DBSCAN et l'analyse hiérarchique ascendante (CAH), ainsi que la stabilité des clusters au fil du temps en utilisant l'Adjusted Rand Index (ARI).



**Merci de
m'avoir écouté
et place aux
questions !**