

# Data Scientist



Présenté par:  
Kchaou Mariem

Encadré par:  
Mr Wilfred Josset

Septembre 2022

# Plan

**INTRODUCTION**

**SOURCE DE DONNÉES**

**SEGMENTATION AVEC L'ANALYSE RFM**

**KMEANS**

**DBSCAN**

**CLUSTERING HIÉRARCHIQUE**

**LA STABILITÉ ET MAINTENANCE DU MODÈLE**

**CONCLUSION**

The logo for olist, featuring the word "olist" in a white, lowercase, sans-serif font, centered on a solid blue rectangular background.

olist

# Contexte

- Toute entreprise aspire à un retour sur investissement le plus rapidement possible.
- Pour atteindre cet objectif pour les sites d'e-commerces, les entreprises préconisent la segmentation pour réduire les ressources à allouer dans le secteur du marketing.
- La segmentation des clients dans notre contexte consiste à découper analytiquement en sous-clients homogènes.
- Ces segments constituent la base des campagnes de communication des équipes de marketing, c'est dire l'importance d'une telle opération.

## **Problématiques et objectifs**

De diverses difficultés se dressent dans la recherche des segments clients, donc :

- Comprendre les différents types d'utilisateurs grâce à leurs données personnelles.
- Rechercher les critères ou variables de mise en évidence des segments pour une utilisation optimale.

Objectifs:

- Fournir aux équipes d'e-commerce une segmentation des clients pour les campagnes de communication.
- Comprendre les différents types d'utilisateurs.
- Fournir une description actionable de la segmentation.
- Faire une proposition de contrat de maintenance.

## SOURCE DE DONNÉES

# Description

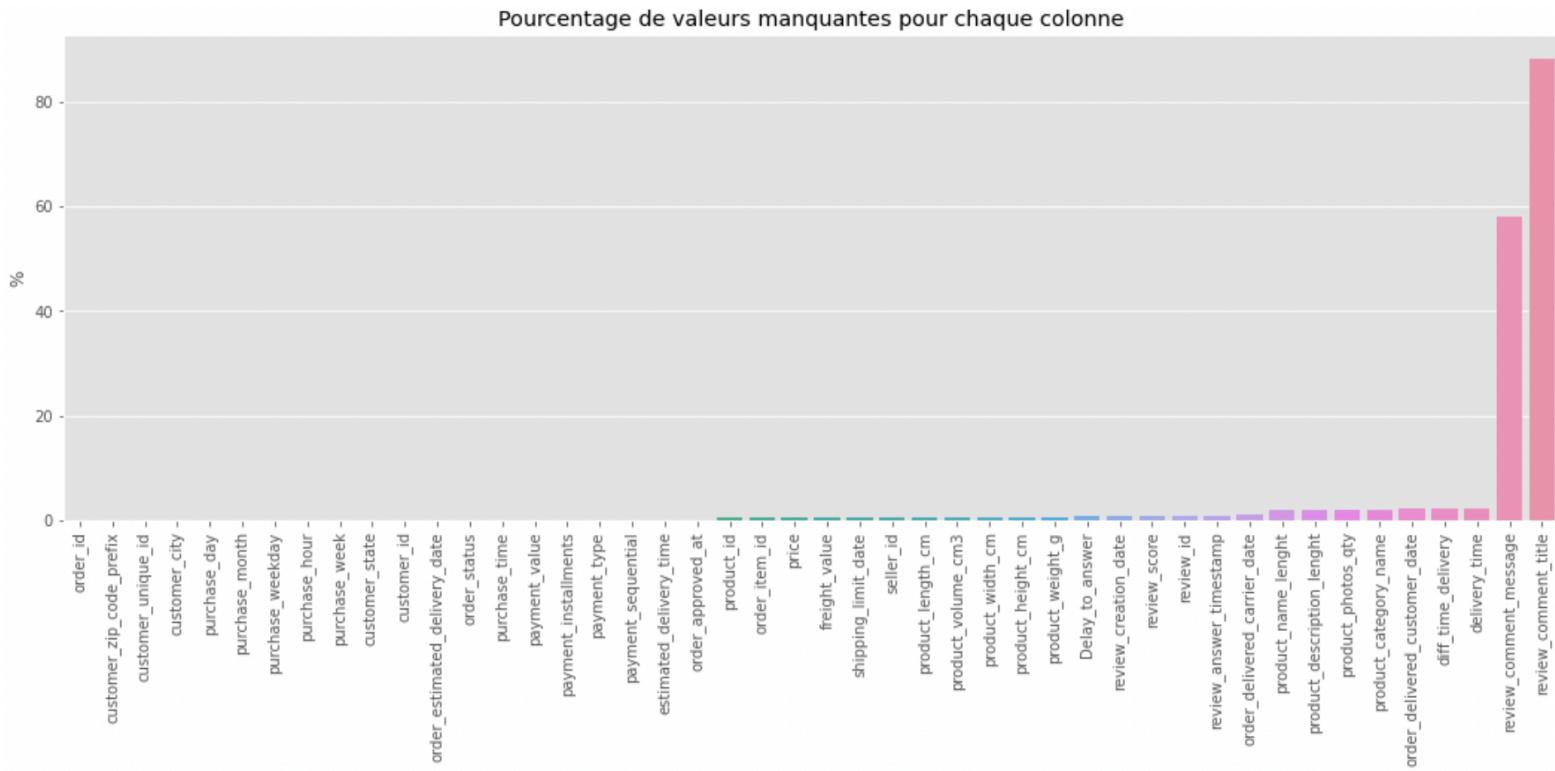
Fichier	Taille	Description
Données géographiques (olist_geolocation_dataset.csv)	1000163 lignes 5 colonnes	Information de localisation en fonction du code postale (latitude, longitude, ville, état)
Données sur les paiements (olist_order_payments_dataset.csv)	103886 lignes 5 colonnes	Information sur le paiement des commandes (nombre de paiement, moyen utilisé, montant)
Données sur les «items» (olist_order_items_dataset.csv)	112650 lignes 7 colonnes	Information sur les ID commande, ID vendeur et ID produit ainsi que des informations sur la commande (prix et date)
Données sur les vendeurs (olist_sellers_dataset.csv)	3095 lignes 4 colonnes	Information sur les vendeurs (localisation et ID vendeur)
Données sur les acheteurs (olist_customers_dataset.csv)	99441 lignes 5 colonnes	Information sur les clients (localisation et ID client)
Données sur les catégories des produits (product_category_name_translation.csv)	71 lignes 2 colonnes	Traduction des catégories de produit du portugais à l'anglais
Données sur l'évaluation des produits par les clients (olist_order_reviews_dataset.csv)	99224 lignes 7 colonnes	Information sur les évaluations des commandes (note, commentaires, date, temps)
Données sur les produits (list_products_dataset.csv)	32951 lignes 9 colonnes	Information sur les produits (type, description, volume, photo,...)
Données sur les achats (olist_orders_dataset.csv)	99441 lignes 8 lignes	Informations sur les commandes (ID client, ID commande, statut, délai de livraison)

## Concaténation des données

- Fusionnement des dataframes sellers et geolocalisation par la colonne "geolocation\_zip\_code\_prefix".
- Fusionnement des dataframes orders et reviews par la colonne commune 'order\_id'.
- Fusionnement des données clients avec les données géographiques
- Fusionnement des dataframes orders et customers par la colonne commune 'customer\_id'.
- Fusionnement des dataframes items et produits par la colonne commune 'product\_id'.
- Fusionnement des orders et des produits par la colonne 'order\_id'.
- Une dataframe contenant 118393 lignes et 46 colonnes.

## SOURCE DE DONNÉES

# Nettoyage des données



## Nettoyage des données

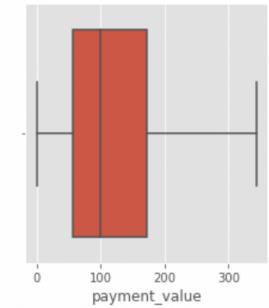
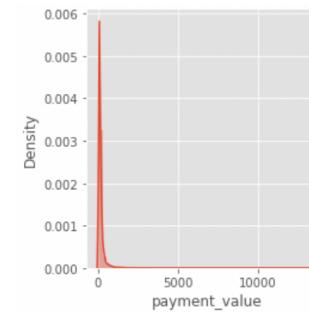
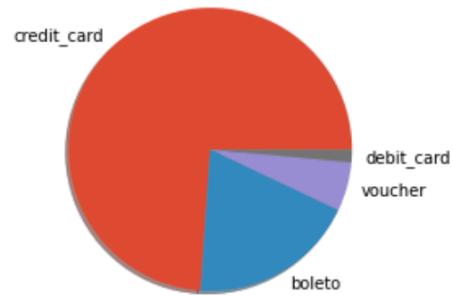
- Suppression des colonnes dont les données sont trop fortement manquantes.
- Remplacement des valeurs manquantes pour chaque colonnes numérique par les moyennes des colonnes.
- Remplacement des valeurs manquantes catégorielles en retypage des catégories en ‘category’.
- Remplacement des valeurs datetime en bonne format.

Le jeu des données compte 44 colonnes et 117416 lignes.

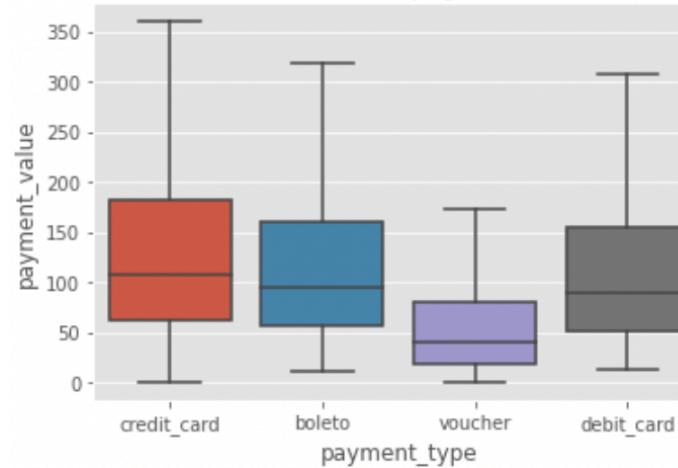
## SOURCE DE DONNÉES

# Exploration des données

Proportion de chaque type de paiement



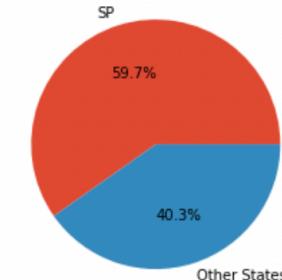
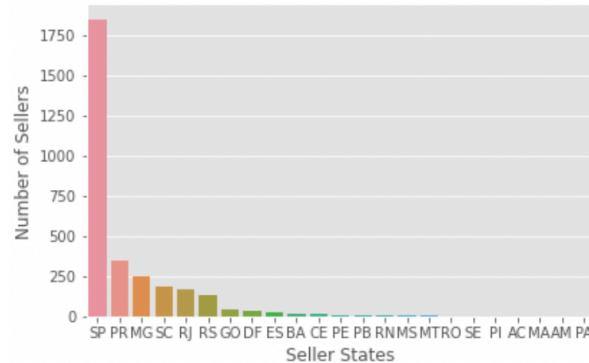
Montant du paiement



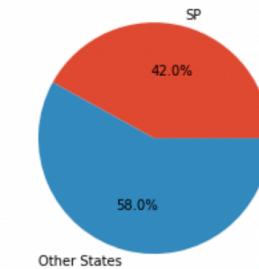
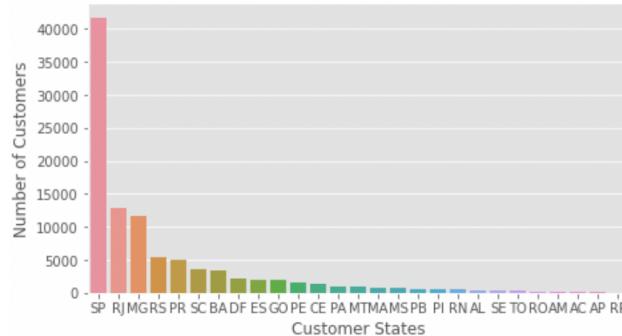
## SOURCE DE DONNÉES

# Exploration des données

Number of Sellers and proportions



Number of Customers and proportions

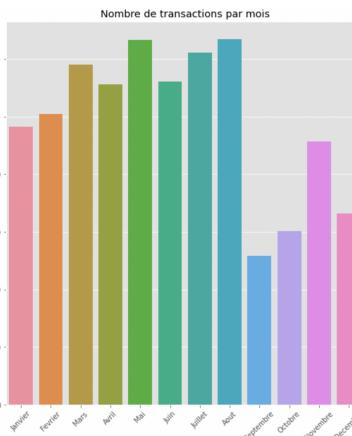
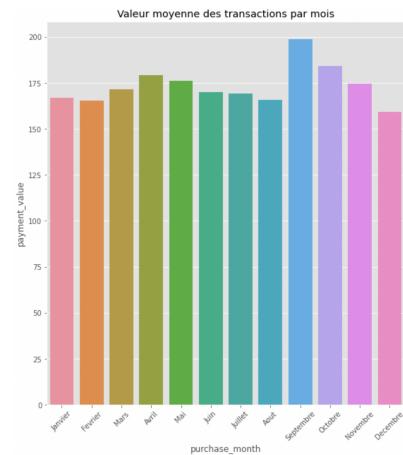
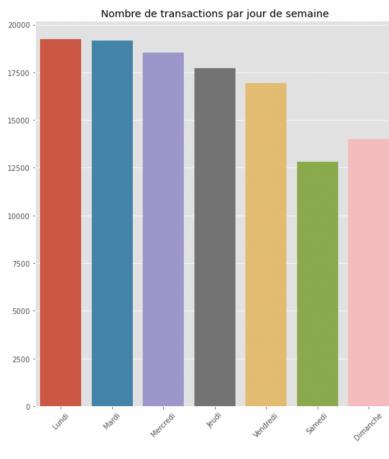
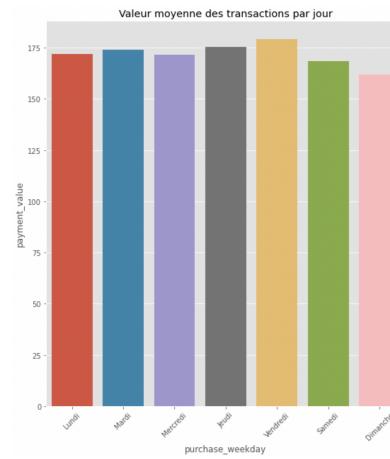
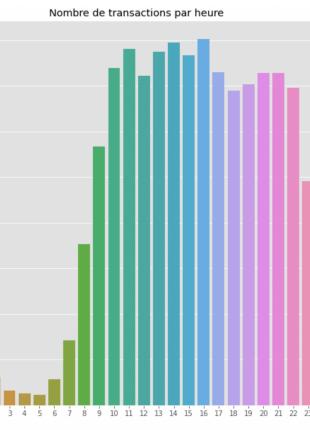
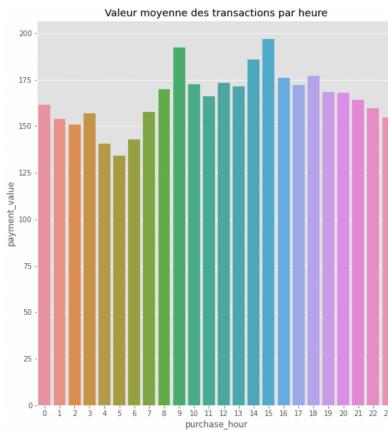


## Exploration des données



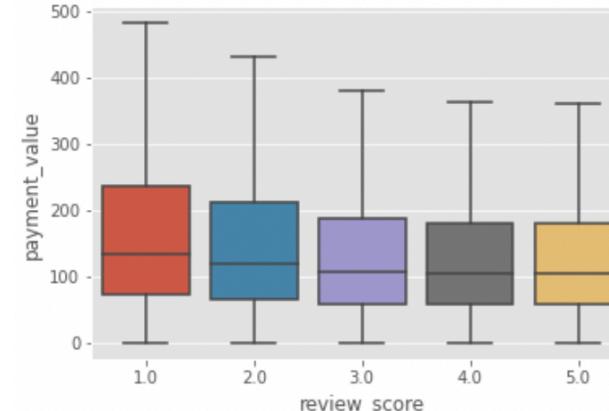
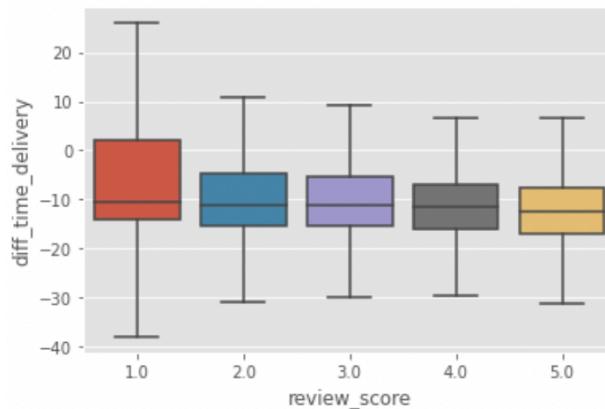
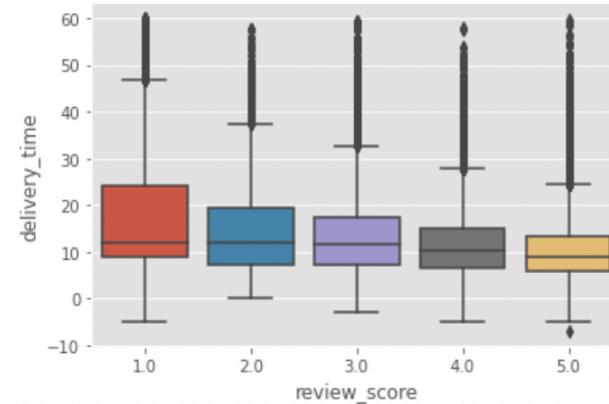
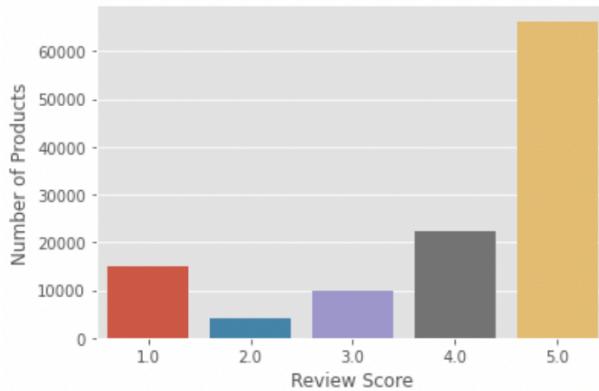
## SOURCE DE DONNÉES

# Exploration des données



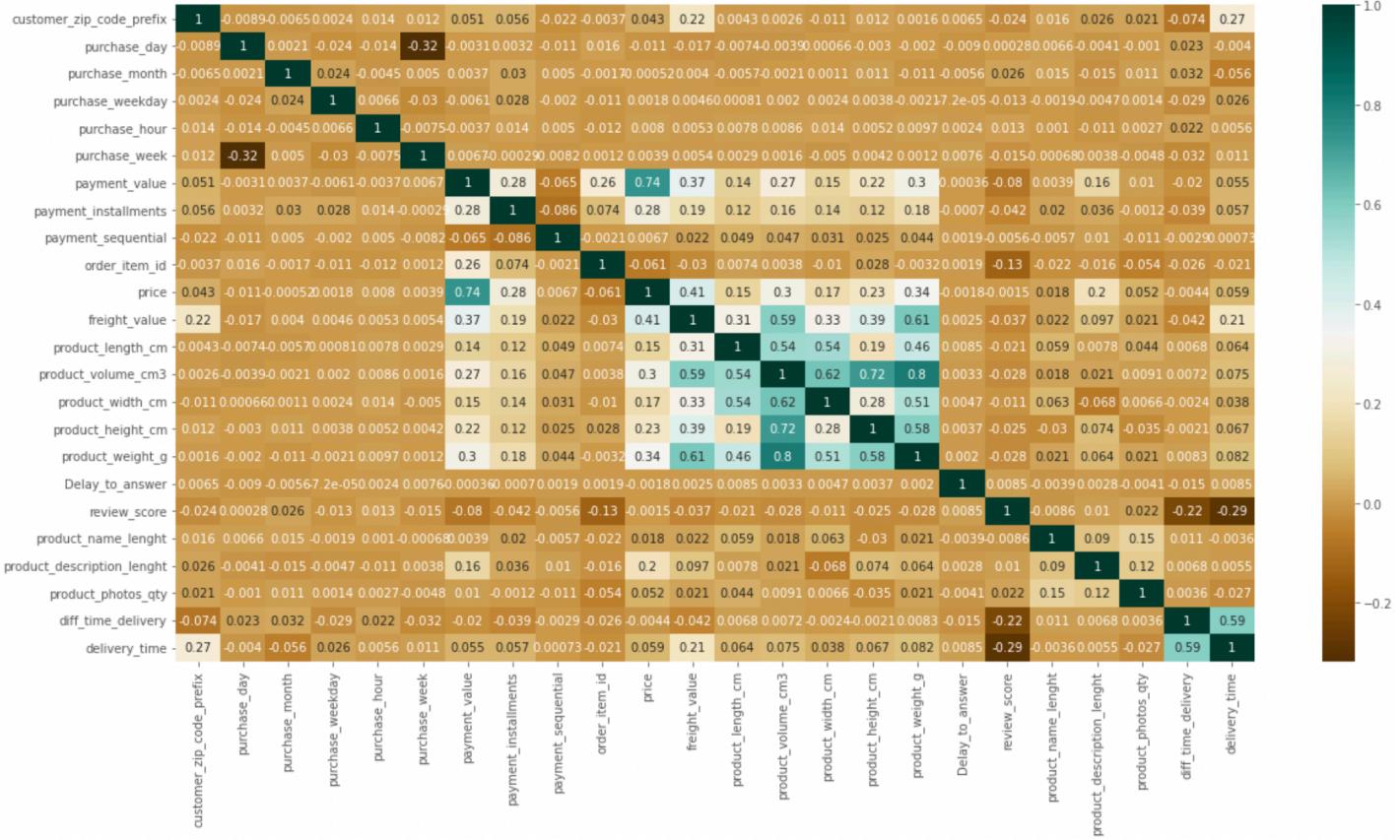
## SOURCE DE DONNÉES

# Exploration des données



## SOURCE DE DONNÉES

# Exploration des données



## SEGMENTATION AVEC L'ANALYSE RFM

RFM est un acronyme pour Récence, Fréquence et Montant.

- Récence indique la date du dernier achat. Il s'agit du nombre de jours depuis la dernière commande d'un client.
- Fréquence est le nombre d'achat sur une période déterminé. Cela peut être 3 mois, 6 mois ou un 1 an. Elle indique la fidélité d'un client, plus sa valeur est élevé et plus le client est engagé.
- Montant est la somme totale qu'un client dépense sur une période donnée.



### Récence

La proximité du dernier achat

Ex : durée depuis le dernier achat



### Fréquence

Référence des achats sur une période

Ex : nombre d'achats sur la dernière année



### Montant

Valeur client sur une période

Ex : Somme de tous les montants d'achat sur la dernière année

## SEGMENTATION AVEC L'ANALYSE RFM

### Le table RFM obtenue

```
# Calcul RFM, entre la date de la dernière commande et il y a 365 jours  
rfm_1 = calculate_rfm(data, period, today)
```

```
# Affichage des 5 premières lignes  
rfm_1.head()
```

	customer_id	Recency	Frequency	Monetary
0	00012a2ce6f8dcda20d059ce98491703	293	1	114.74
1	000379cdec625522490c315e70c7a9fb	154	1	107.01
2	000419c5494106c306a97b5635748086	185	1	49.40
3	00046a560d407e99b969756e0b10f282	259	1	166.59
4	00050bf6e01e69d5c0fd612f1bcfb69c	351	1	85.23

## SEGMENTATION AVEC L'ANALYSE RFM

```
quantiles = rfm_1.quantile(q=[0.25,0.5,0.75])
quantiles
```

	Recency	Frequency	Monetary
0.25	94.0	1.0	63.1625
0.50	178.0	1.0	111.1200
0.75	265.0	1.0	196.5425

Attribuer les valeurs selon les quartiles.

```
RFM_Segment.head()
```

	customer_id	Recency	Frequency	Monetary	R	F	M	RFMClass
0	00012a2ce6f8dcda20d059ce98491703	293	1	114.74	1	1	3	113
1	000379cdec625522490c315e70c7a9fb	154	1	107.01	3	1	2	312
2	000419c5494106c306a97b5635748086	185	1	49.40	2	1	1	211
3	00046a560d407e99b969756e0b10f282	259	1	166.59	2	1	3	213
4	00050bf6e01e69d5c0fd612f1bcfb69c	351	1	85.23	1	1	2	112

## **SEGMENTATION AVEC L'ANALYSE RFM**

Chaque client est caractérisé en fonction de la note pour chaque variable R, F M qui le caractérise

les segments sont définis :

- Champions : achats récents, achète souvent et dépense beaucoup
- Loyal Customers : achètent régulièrement, sensibles aux promotions
- Potential Loyalist : nouveaux clients avec fréquence d'achat moyenne
- Recent Customers : achats récents, mais peu fréquents
- Promising : nouveaux acheteurs, mais peu dépensiers
- Need Attention : dépenses moyennes, moyennement fréquemment, achat moyennement récent.
- About to Sleep : client qui tendent à disparaître
- Can't Lose Them : achat régulière, besoin de les faire revenir
- At Risk : dépensaient beaucoup, achetaient souvent mais il y a longtemps
- Lost : peu dépensiers, peu fréquemment, a acheté il y a un moment
- Others : pas catégorisante

# SEGMENTATION AVEC L'ANALYSE RFM

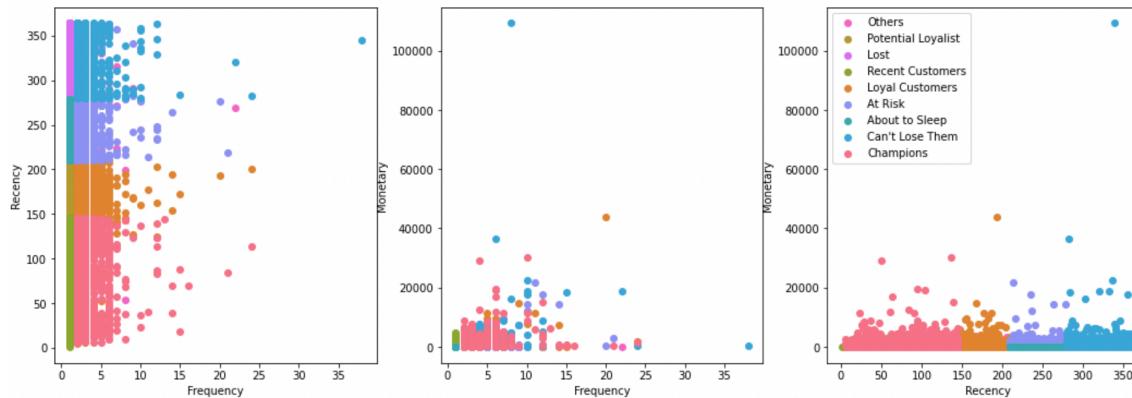
```
# Affiche les 5 premières lignes  
rfm_table_1.head()
```

	customer_id	Recency	Frequency	Monetary	R	F	M	RFMClass	RFM Score
0	00012a2ce6f8dcda20d059ce98491703	293	1	114.74	1	1	3	113	Others
1	000379cdec625522490c315e70c7a9fb	154	1	107.01	3	1	2	312	Potential Loyalist
2	000419c5494106c306a97b5635748086	185	1	49.40	2	1	1	211	About to Sleep
3	00046a560d407e99b969756e0b10f282	259	1	166.59	2	1	3	213	Others
4	00050bf6e01e69d5c0fd612f1bcfb69c	351	1	85.23	1	1	2	112	Lost

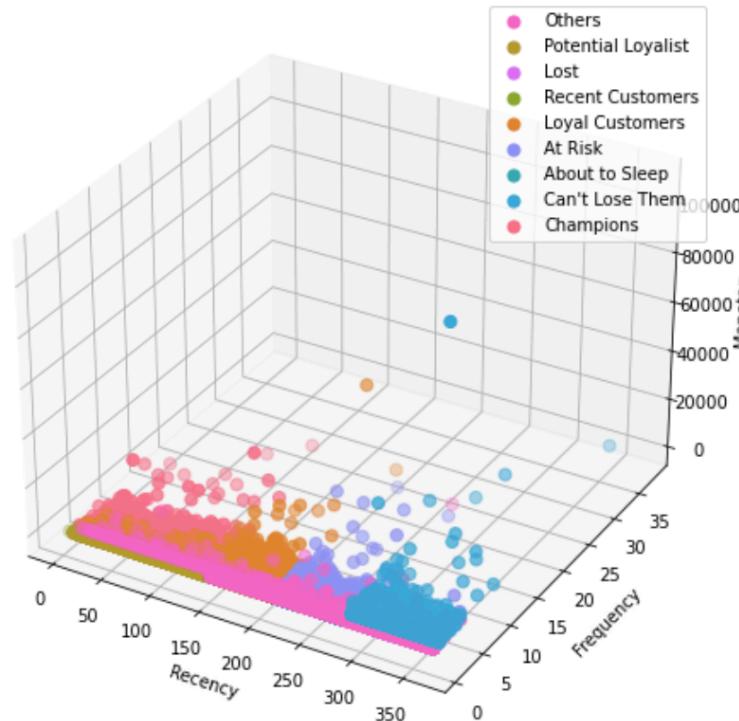
```
# Liste des fonctions d'agrégation  
func = ['count', 'min', 'mean', 'median', 'max', 'std']  
# Regroupement des données  
qtd_reco = rfm_table_1.groupby(['RFM Score'])  
# Agrégation des données par différentes fonctions  
summery_table = qtd_reco[['Recency', 'Frequency', 'Monetary']].agg(func)  
# Affichage de la table de résumé  
summery_table
```

RFM Score	Recency					Frequency					Monetary							
	count	min	mean	median	max	std	count	min	mean	median	max	std	count	min	mean	median	max	std
About to Sleep	9175	179	218.449700	217.0	265	24.232992	9175	1	1.000000	1.0	1	0.000000	9175	13.78	63.539973	62.510	111.10	24.515768
At Risk	2863	179	237.474328	228.0	364	45.453218	2863	2	2.427174	2.0	24	1.261227	2863	63.17	495.072078	242.400	44048.00	1255.839520
Can't Lose Them	1727	266	305.959467	299.0	364	28.083978	1727	2	2.678054	2.0	38	1.711016	1727	196.96	884.533185	442.880	109312.64	3099.093516
Champions	1555	5	51.713826	48.0	94	25.003908	1555	2	2.621222	2.0	21	1.433624	1555	196.68	789.492431	456.760	29099.52	1422.717748
Lost	8996	266	307.248110	300.0	364	29.363256	8996	1	1.000000	1.0	1	0.000000	8996	10.07	64.535883	64.000	111.12	23.776142
Loyal Customers	2826	6	120.875088	127.0	178	41.554855	2826	2	2.456476	2.0	24	1.224983	2826	63.28	546.891023	268.160	30186.00	1139.944814
Others	17909	6	238.106594	238.0	364	69.756818	17909	1	1.044726	1.0	22	0.326854	17909	9.59	275.276445	196.920	4175.26	269.982077
Potential Loyalist	26609	1	92.919614	94.0	178	50.510769	26609	1	1.000000	1.0	1	0.000000	26609	10.89	91.554733	83.870	196.52	46.221426
Recent Customers	3106	6	50.347714	47.0	94	24.665782	3106	1	1.000000	1.0	1	0.000000	3106	196.55	415.965821	286.015	4681.78	364.269643

## SEGMENTATION AVEC L'ANALYSE RFM



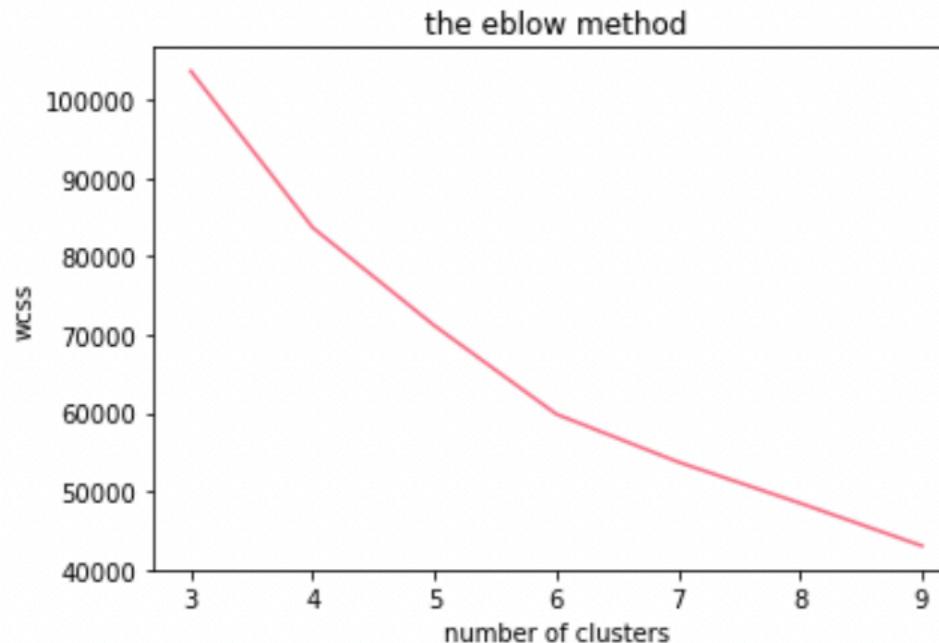
Représentation 3D des différents individus dans chaque segment



# Méthodologie

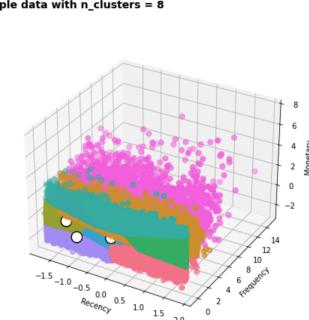
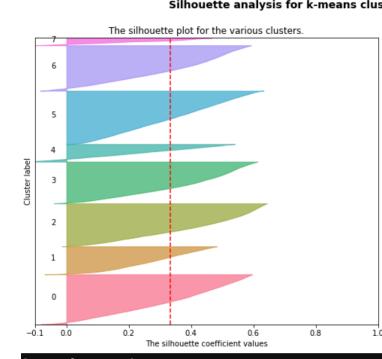
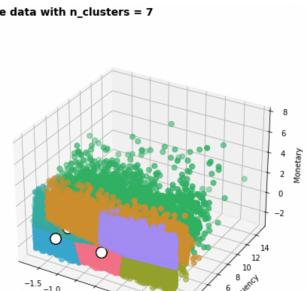
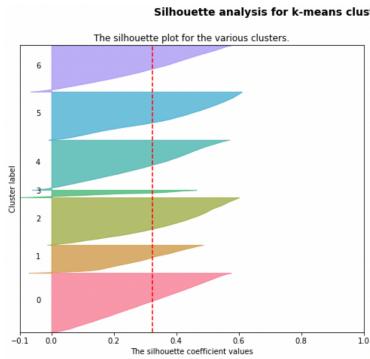
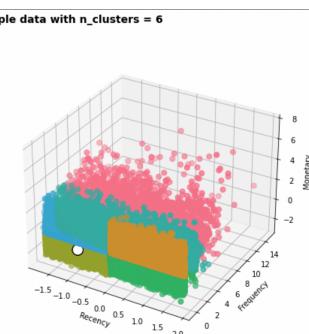
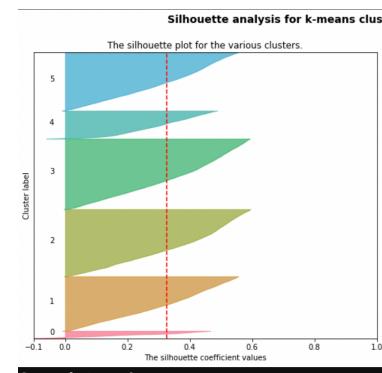
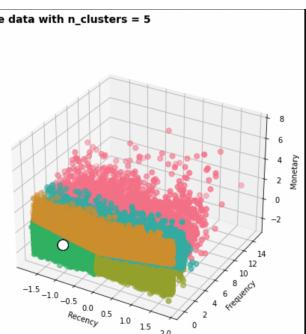
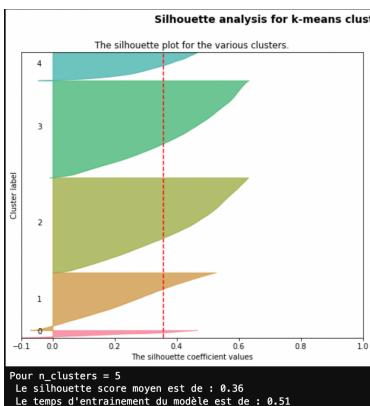
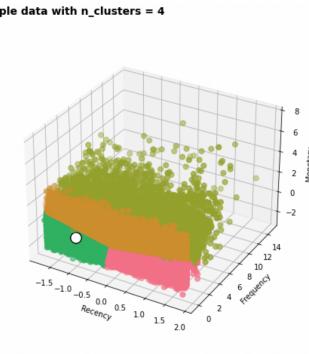
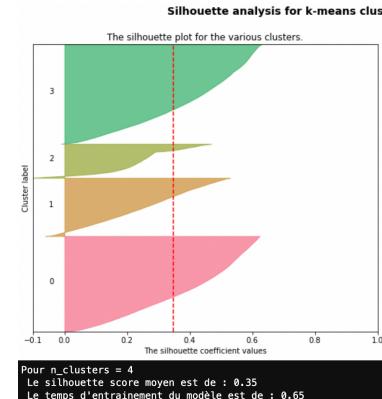
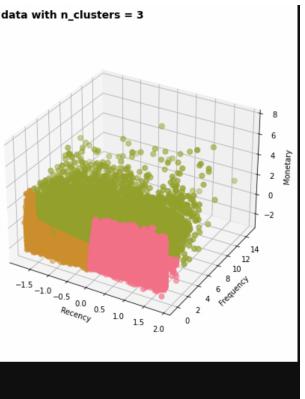
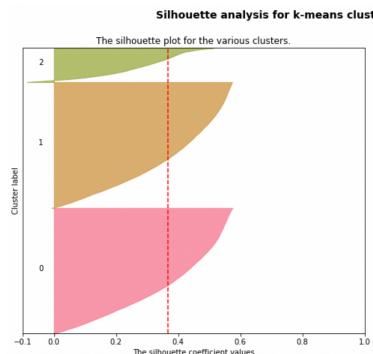
- Utiliser les variables les plus pertinentes (customer\_id, purchase\_time, payment\_value).
- Normalisation des données avec StandardScaler().
- Clusterisation des données par k-means.
- Choisir le nombre de clusters.
- Etiquetter les clusters

## KMEANS



Le nombre de clusters = 6

# KMEANS



## KMEANS

Descriptifs du clustering des consommateurs :

- Top Clients ++ : Il s'agit des grands comptes, les individus dépensent très régulièrement des grandes quantités d'argent
- Good Clients : Ils arrivent après les grandes comptes, les sommes d'argent dépensés sont conséquentes et la fréquence élevé
- Potential Loyalist : nouveaux clients avec fréquence d'achat moyenne
- Recent Customers : achats récents, mais peu fréquents
- Lost or About to Sleep: clients qui n'ont pas effectués d'achat cette année ou qui tendent à disparaître
- Others : pas catégorisante

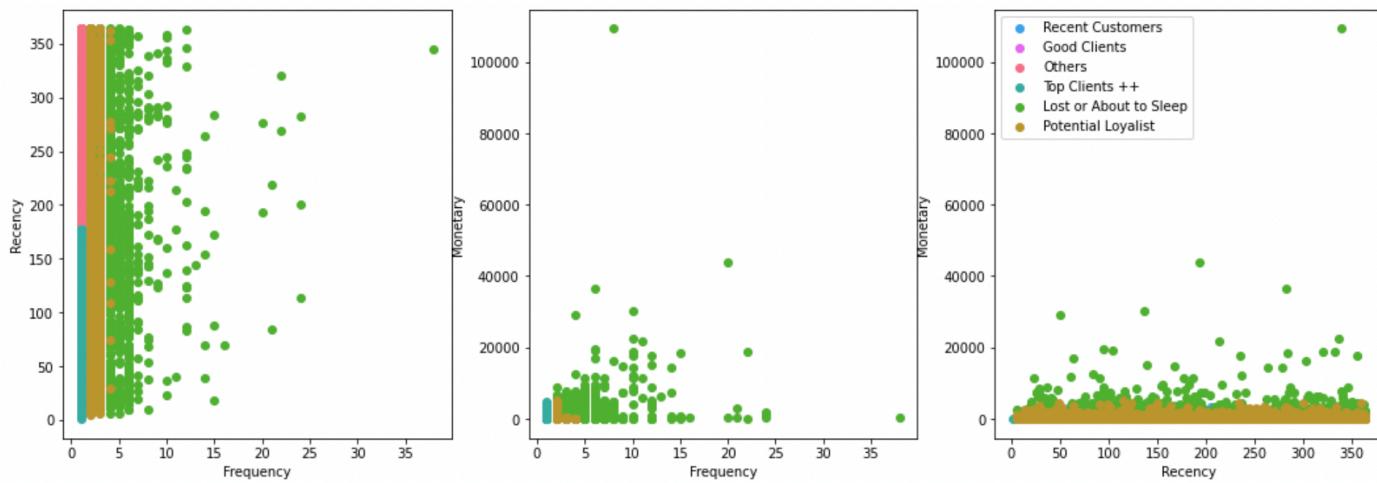
```
rfm_table_1.head()
```

	customer_id	Recency	Frequency	Monetary	R	F	M	RFM Score	clusters
0	00012a2ce6f8dcda20d059ce98491703	293	1	114.74	1	1	3	Others	Recent Customers
1	000379cdec625522490c315e70c7a9fb	154	1	107.01	3	1	3	Potential Loyalist	Good Clients
2	000419c5494106c306a97b5635748086	185	1	49.40	3	1	1	Potential Loyalist	Others
3	00046a560d407e99b969756e0b10f282	259	1	166.59	2	1	4	Others	Recent Customers
4	00050bf6e01e69d5c0fd612f1bcfb69c	351	1	85.23	1	1	2	Lost	Others

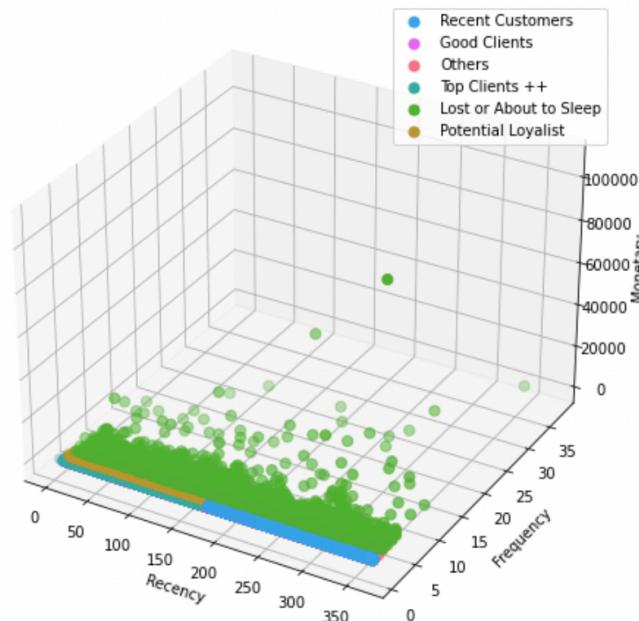
# KMEANS

clusters	Recency						Frequency						Monetary																								
	count		min		mean		median		max		std		count		min		mean		median		max		std														
	Good Clients	17654	5	91.829897	92.0	178	50.508587	17654	1	1.007590	1.0	2	0.086794	17654	9.59	63.289168	62.25	110.63	24.219252	Lost or About to Sleep	1939	6	187.804023	188.0	364	98.500249	1939	2	4.308922	4.0	38	2.238830	1939	19.35	1582.182955	836.20	109312.64
Others	18417	177	262.037574	264.0	364	52.234250	18417	1	1.009285	1.0	2	0.095912	18417	10.07	63.711634	62.88	110.90	24.082939	Potential Loyalist	7306	5	180.253490	181.0	364	96.349747	7306	2	2.054065	2.0	4	0.233312	7306	17.38	364.557038	270.00	5277.28	372.862734
Recent Customers	14296	178	262.133114	265.0	364	52.096282	14296	1	1.000000	1.0	1	0.000000	14296	110.55	251.359617	177.16	3826.80	232.162805	Top Clients ++	15154	1	93.443843	94.0	177	49.557793	15154	1	1.000000	1.0	1	0.000000	15154	109.21	258.965743	176.04	4681.78	271.828559

# KMEANS



Représentation 3D des différents individus dans chaque segment



## Méthodologie

- Utilisation les même variables donnant un résultat optimal pour Kmeans (customer\_id, purchase\_time, payment\_value).
- Paramètres à optimiser: rayon de recherche autour de chaque point (avec la distance euclidienne) et le nombre maximal de points dans le rayon nécessaires pour qu'un point soit considéré dans un cluster
- Valeurs testées: epsilon est la distance maximale entre deux échantillons pour qu'un seul soit considéré comme au voisinage de l'autre, cette valeur étant 1.05 et nous testons les valeurs 0.25, 0.5, 0.75, 1. Min\_sample est un nombre trop faible ne devrait pas nous permettre d'avoir des clusters de taille satisfaisante, nous utiliserons 50 et 1500.



L'algorithme DBSCAN nécessitait une consommation de mémoire vive trop importante pour que je puisse le lancer sur le jeu de données complet avec mon ordinateur. J'ai donc effectué le travail sur la moitié du jeu de données (avec sélection aléatoire des points)

## DBSCAN

eps= 1.05, n\_simple= 300

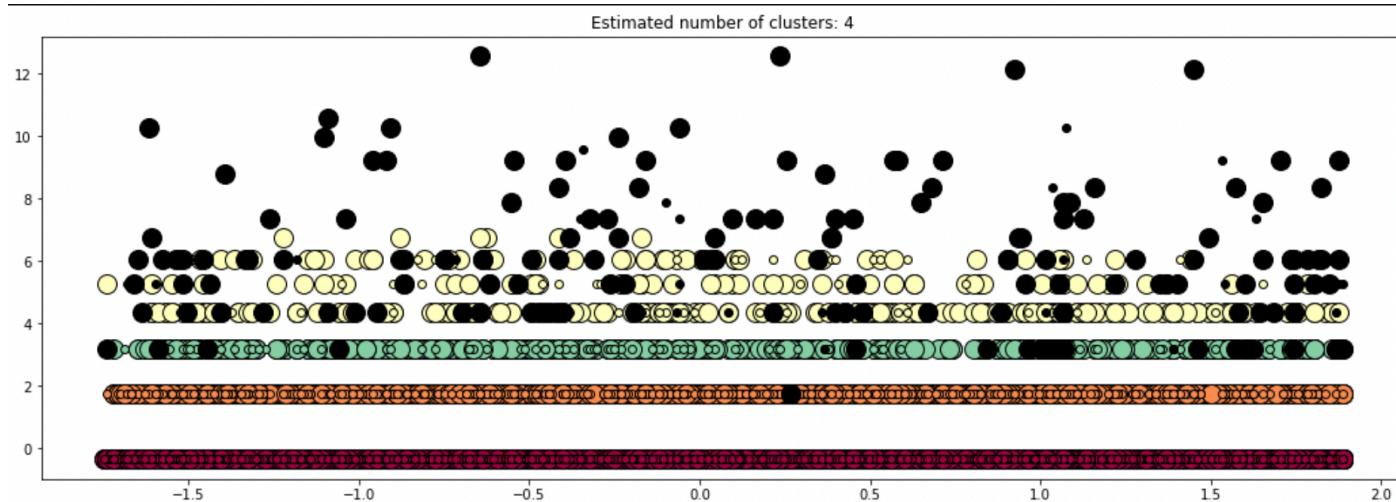
```
Estimated number of clusters: 2
Estimated number of noise points: 1201
Silhouette Coefficient: 0.4142
```

eps= 1.05, n\_simple=50

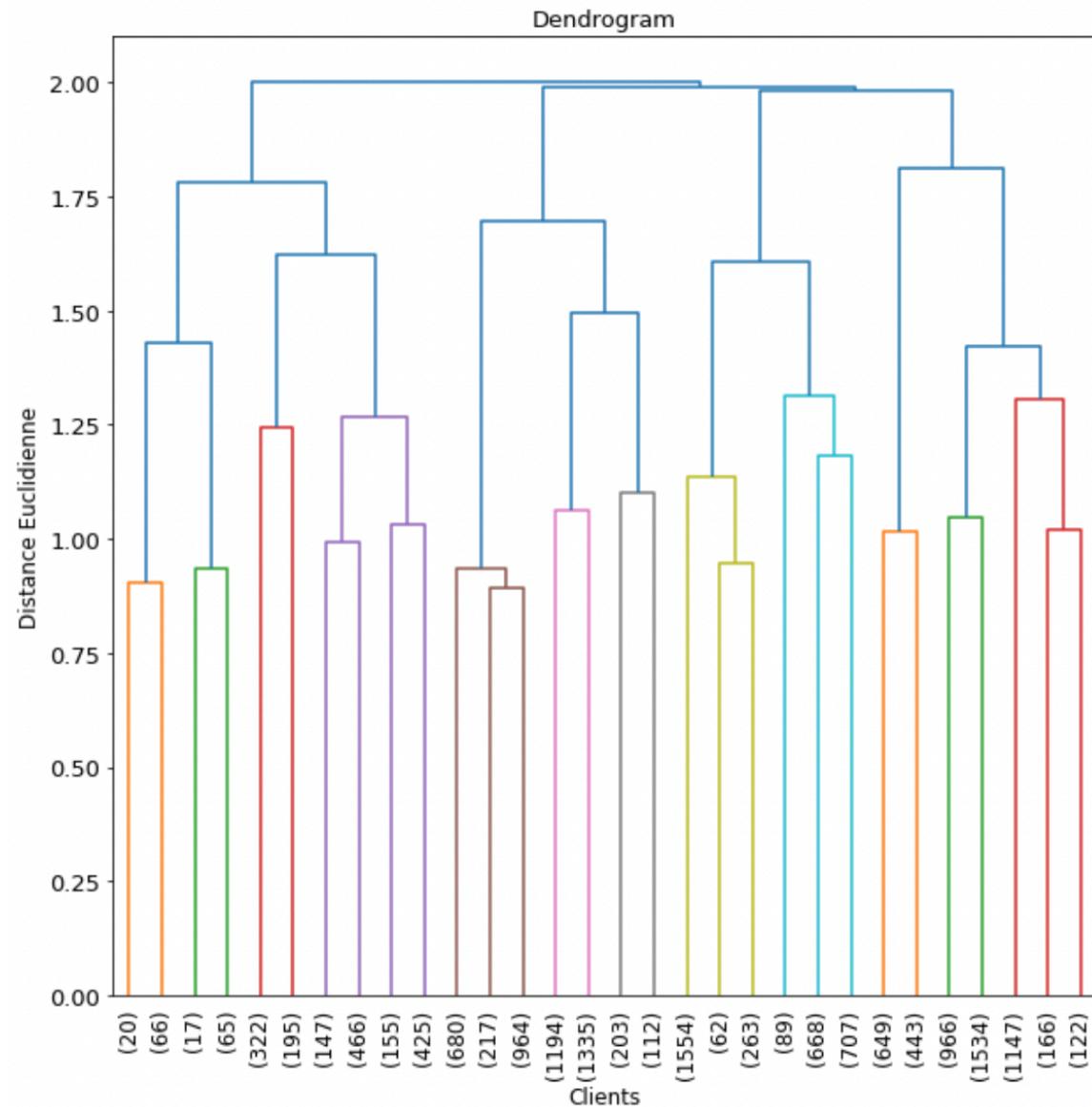
```
Estimated number of clusters: 4
Estimated number of noise points: 176
Silhouette Coefficient: 0.4053
```

## DBSCAN

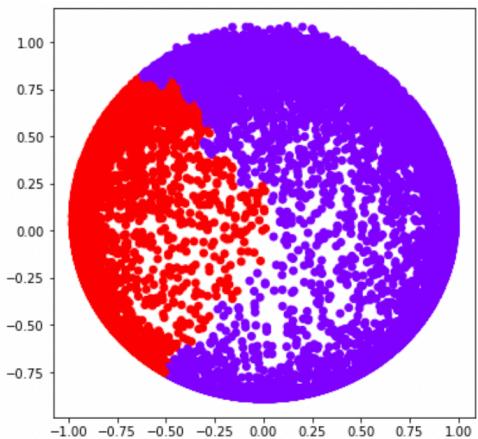
- Cluster 1: 32558 clients (87,09%)
- Cluster 2: 3649 clients (9,76%)
- Cluster 3: 601 clients (1,61%)
- Cluster 4: 399 clients (1,07%)
- Les autres clients (0,47%) sont classifiées en tant que bruit (cluster -1)



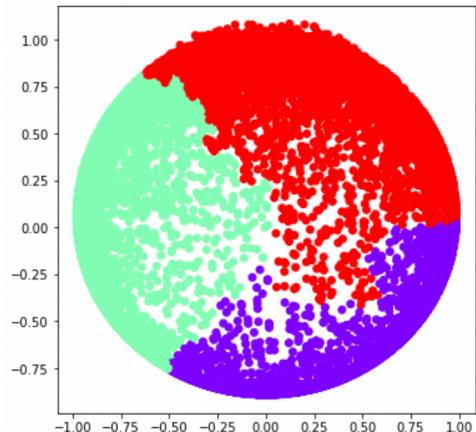
## CLUSTERING HIÉRARCHIQUE



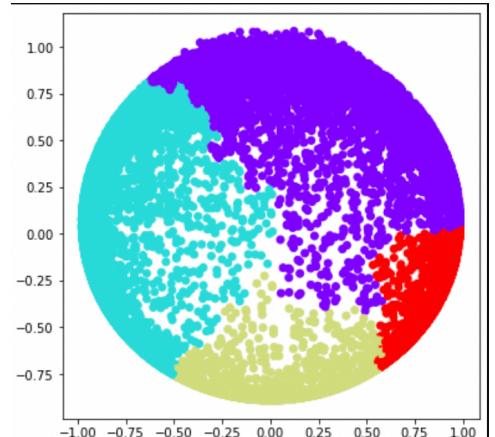
## CLUSTERING HIÉRARCHIQUE



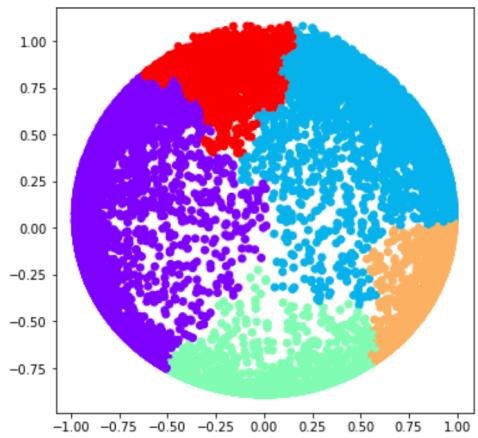
$n_{clusters}=2$



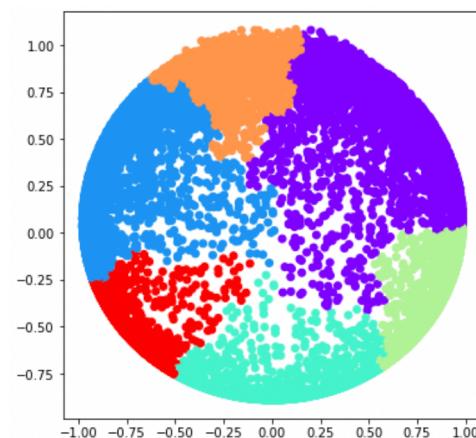
$n_{clusters}=3$



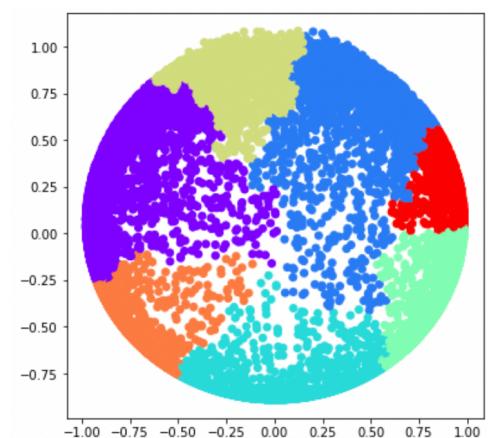
$n_{clusters}=4$



$n_{clusters}=5$

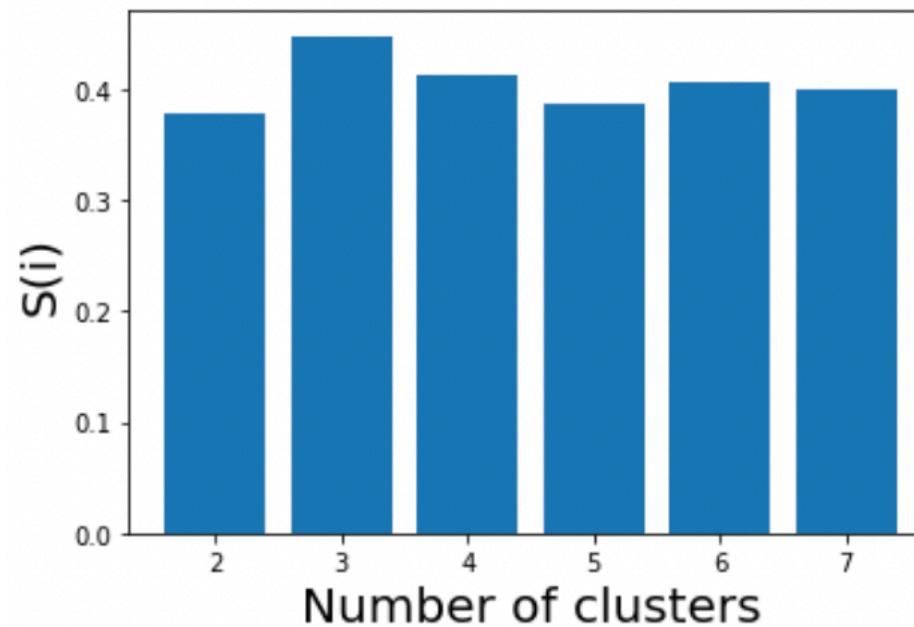


$n_{clusters}=6$

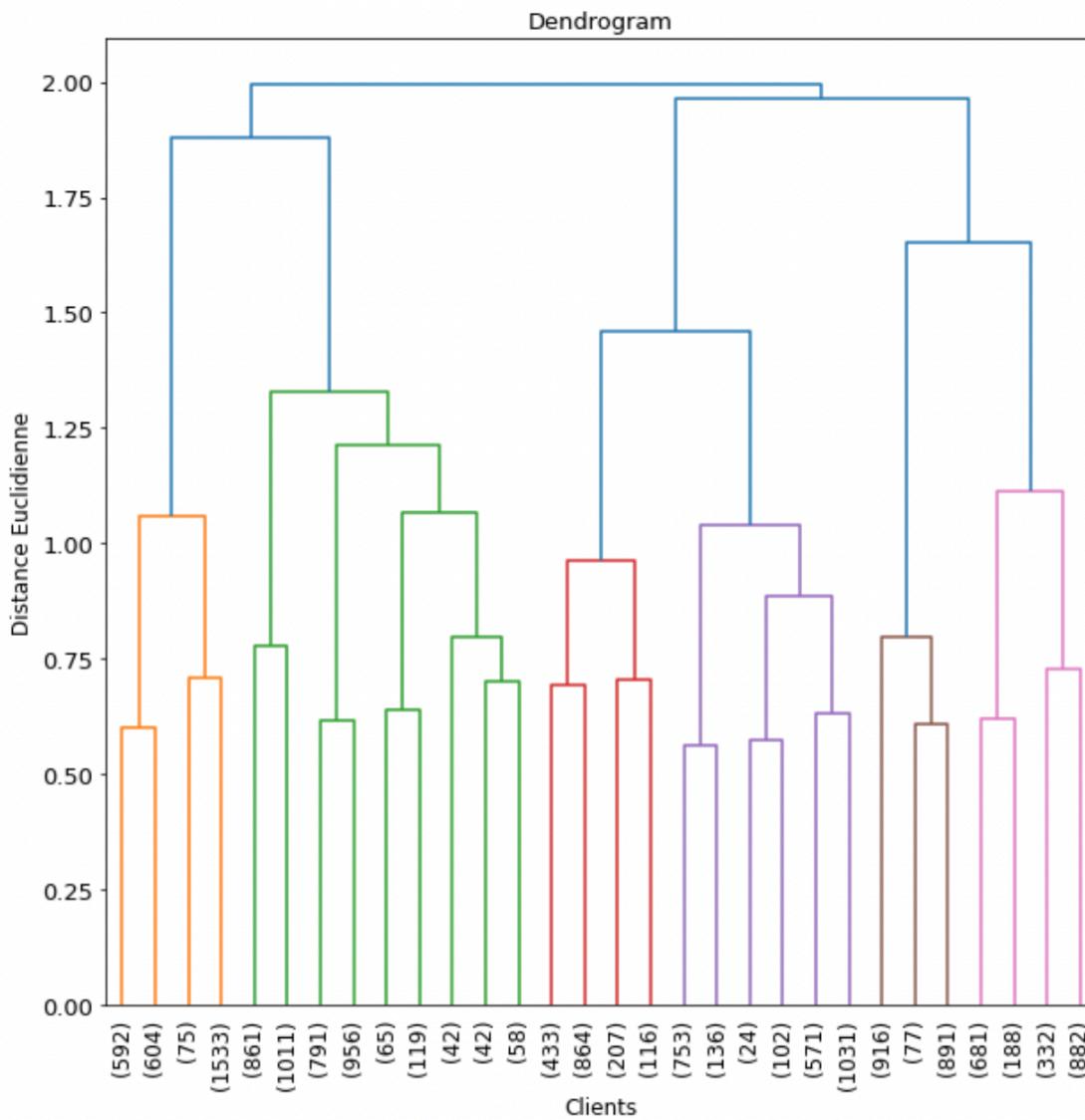


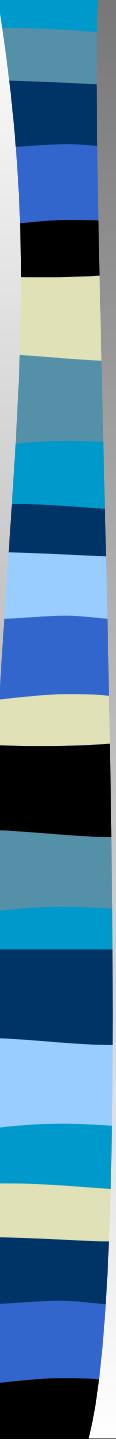
$n_{clusters}=7$

## CLUSTERING HIÉRARCHIQUE



## CLUSTERING HIÉRARCHIQUE

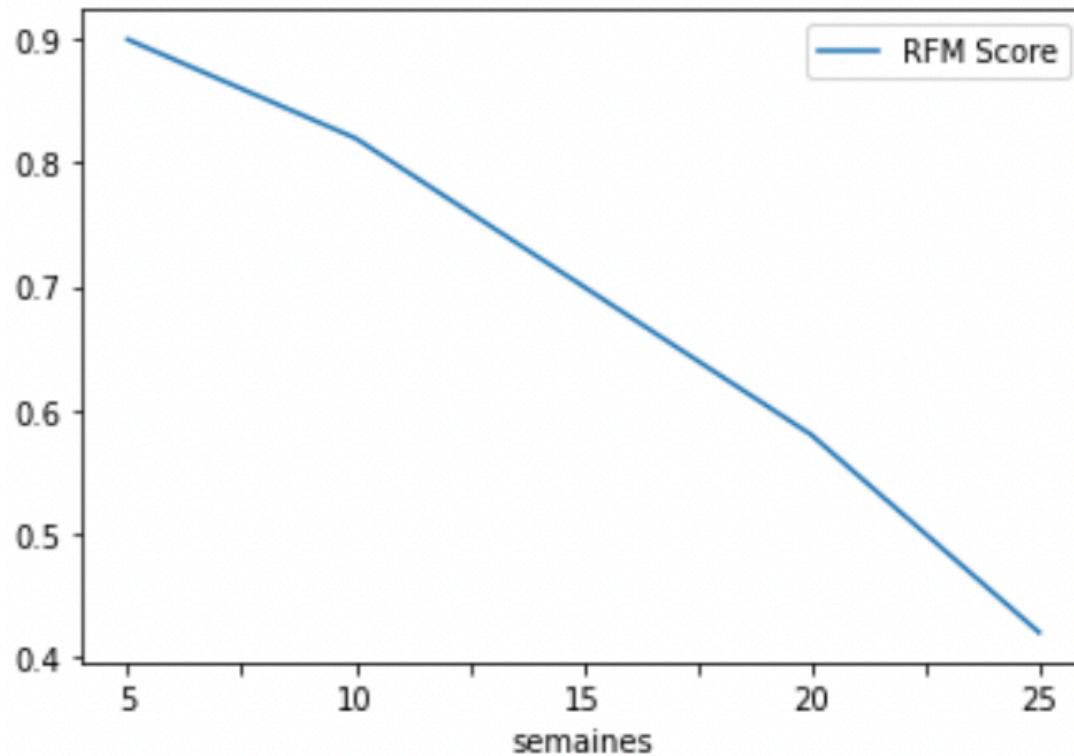




## LA STABILITÉ ET MAINTENANCE DU MODÈLE

- Calculer la durée à partir de laquelle un modèle de clustering devient obsolète considérant l'arrivée de nouveaux clients ou nouvelles commandes effectuées par des clients existants.
- Nous allons essayer d'estimer la durée à partir de laquelle les prédictions de cet algorithme ne sont plus pertinente et nécessite entraîner de nouveau algorithme sur les nouvelles données.
- Nous allons utiliser l'adjusted rand score qui permet de comparer deux classifications sur le même jeu de données.
- Nous allons estimer que si le score est inférieur à 0,8 alors notre algorithme initial ne permet pas d'effectuer des prédictions pertinentes.
- Nous entraînons un modèle initial à un certain jours (365 jours) sur la base des données disponibles.
- 30 jours plus tard le jeu de données va être modifié avec l'intégration de nouvelles commandes (on va faire effectuer une prédition avec le modèle initial, entraîner un nouveau modèle à ce nouveau jeu de données et comparer les prédictions de deux modèles avec adjusted rand score).
- Répéter l'opération jusqu'à obtenir le seuil du score ARI inférieur à 0,8

## LA STABILITÉ ET MAINTENANCE DU MODÈLE



## LA STABILITÉ ET MAINTENANCE DU MODÈLE

Evolution temporelle des clients dans chaque segment RFM



## LA STABILITÉ ET MAINTENANCE DU MODÈLE

KMeans:

Le silhouette score=0,33

Nombre de clusters=6

Algorithme rapide dans l'exécution.

Score moyens mais segmentation intéressante d'un point de vue métier.

DBSCAN:

Le silhouette score=0,40

Nombre de clusters=4

Consommation de mémoire vive trop importante nécessite d'utiliser seulement la moitié du jeu de données.  
Un cluster dispose de trop Peu clients (1,07%)

Clustering hiérarchique:

Le silhouette score=0,41

Nombre de clusters=6

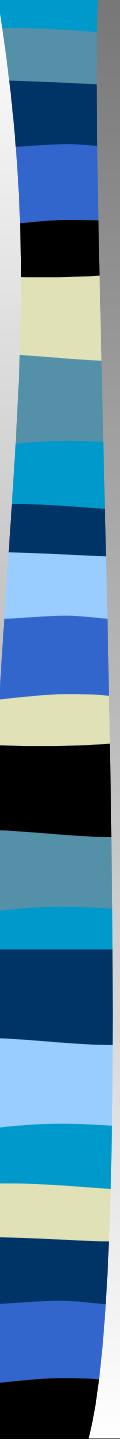
Consommation de mémoire vive trop importante nécessite de faire réduction de dimension avec ACP

Bien que ses scores soient bons l'algorithme DBSCAN ne permet pas d'effectuer une segmentation satisfaisante pour une interprétation métier avec ce jeu de données. Le clustering hiérarchique et le KMeans permettent une segmentation intéressante tant que KMeans affiche des temps de calcul inférieur et permet d'utiliser l'intégralité du jeu de données: **nous retenons donc le KMeans**

## **CONCLUSION**

Nous avons réussi à créer des segments actionnable à l'aide d'algorithme de machine learning en combinaison avec des techniques de marketing.

- La segmentation peut être améliorée par :
  - Création d'une macro-segmentation par type de produit.
  - Ajout d'information sur la réactivité de clients vis-à-vis des promotions.
  - Analyse d'évolution de segments dans le temps
- Selon les attentes de l'équipe marketing (quel type de clients cibler, quels sont les moyens,...)
- Eléments à préciser pour le contrat de maintenance:
  - Nouvelles features/ clients ayant acheté plusieurs articles.
  - Caractérisation dans le détail des produits des champs textuels.
  - Données plus précises sur les clients: âge, sexe,..



*Merci pour votre  
attention*

*Questions*

