Segmentation des clients d'un site d'e-commerce

Sommaire

Segmentation des clients d'un site d'e-commerce

- **Présentation de la problematique**

- **Analyse et visualisation**

- **Prévisions**

- Prévision de la valeur client à long terme et de la segmentation. Une description actionable de la segmentation et de sa logique sous-jacente pour une utilisation optimale.

- **Évaluation de la stabilité du modèle dans le temps,**

de la périodicité du nécessaire réapprentissage du modèle en fonction de l'arrivée de nouvelles données. Une proposition de contrat de maintenance, (fréquence à laquelle la segmentation doit être mise à jour pour rester pertinente), basée sur une analyse de la stabilité des segments au cours du temps.

Présentation

Olist, une entreprise brésilienne qui propose une solution de vente sur les marketplaces en ligne. La solution Olist se compose de trois aspects : logiciel, contrats avec les marchés principaux et échange de réputation. L'entreprise s'intéresse à la segmentation client qu'elle pourrait utiliser au quotidien pour ses campagnes de communication. Pour cette mission, Olist a fourni une base de données anonyme, qui se trouve au lien sur le site https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce

1. What is Olist?

Mission - Segmentation des clients

1) Comprendre **les différents types d'utilisateurs** à travers leur comportement et leurs données personnelles;

2)**La description** efficace **de la segmentation** de la clientèle et de sa logique de base;

3)Une proposition de contrat de maintenance basée sur une analyse de la stabilité de la segmentation dans le temps.

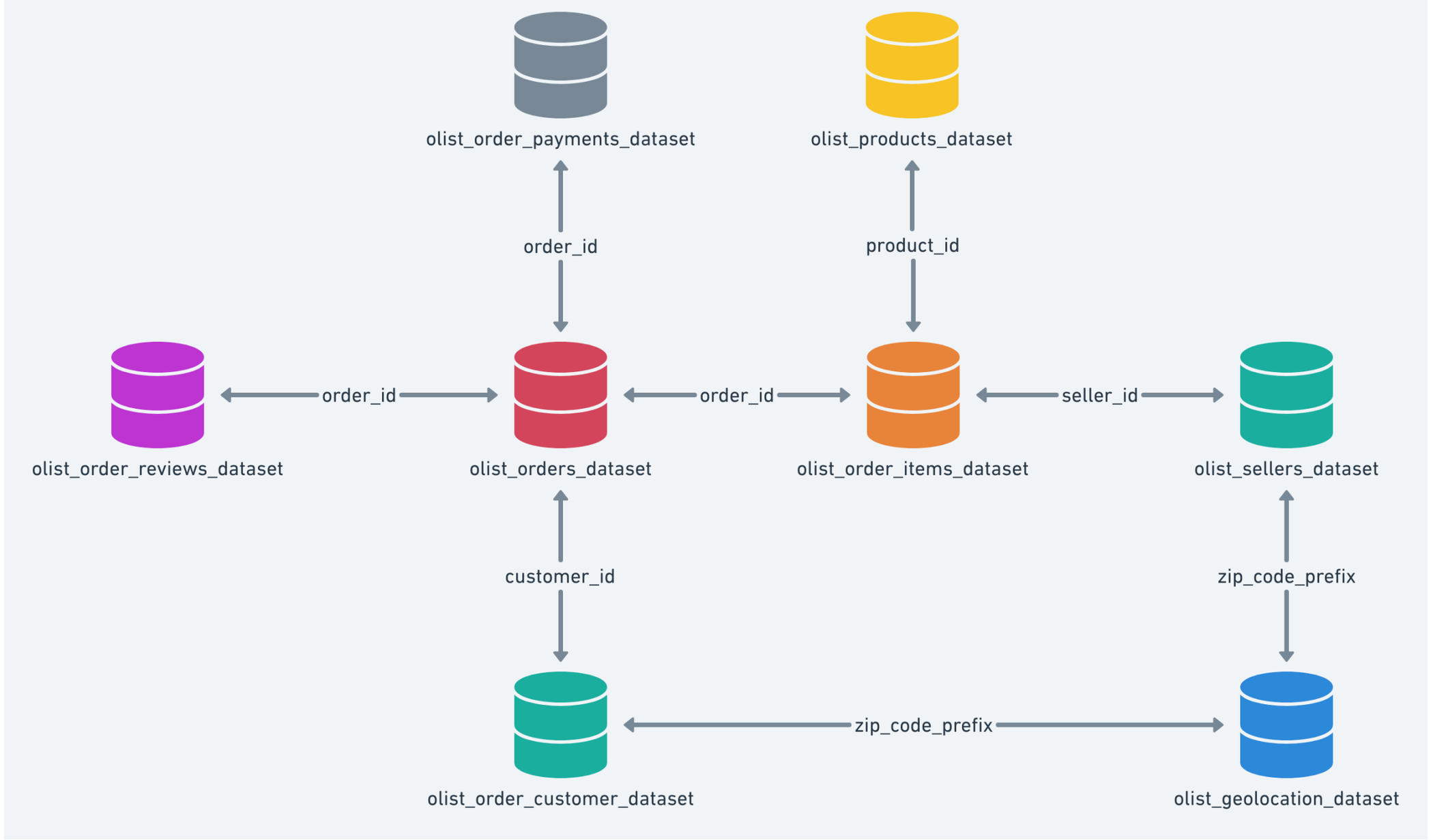
Source de données

- anonymisées
- période limitée de 18 mois
- 9 fichiers CSV à intégrer

Interprétation du problématique

Les démarches

- Nettoyage fusion de données par client
- Analyse exploratoire des dimensions disponibles
- Feature engineering: Sélection / création d'indicateurs de comportement
- Modélisation : Segmentation des clients
- Interprétation: actions à prendre
- Evaluation de la stabilité de segmentation



La base de données- un ensemble de données de commerce électronique brésilien accessible au public sur les commandes passées sur la boutique Olist. L'ensemble de données contient des informations sur 115 000 commandes de 09.2016 à 08.2018 passées sur plusieurs marchés au Brésil. Ses fonctionnalités permettent de visualiser les commandes à partir de plusieurs dimensions, du statut de la commande, du prix, des performances de paiement et de livraison à l'emplacement du client, aux attributs du produit et enfin aux avis des clients. Un ensemble de données de géolocalisation qui relie les codes postaux brésiliens aux coordonnées lat/long est également inclus. Cet ensemble de données contient neuf tables reliées par des attributs communs.

La plateforme Olist - tables

Clients

- customers
- geolocation
- orders

Commandes

- orders
- order reviews
- order payments
- order items

Attributs des produits

- order items
- sellers
- product_category
- products
- geolocation

Customers:

customer_id - **Le** clé de jeu de données de commande. Chaque commande a un identifiant client unique.

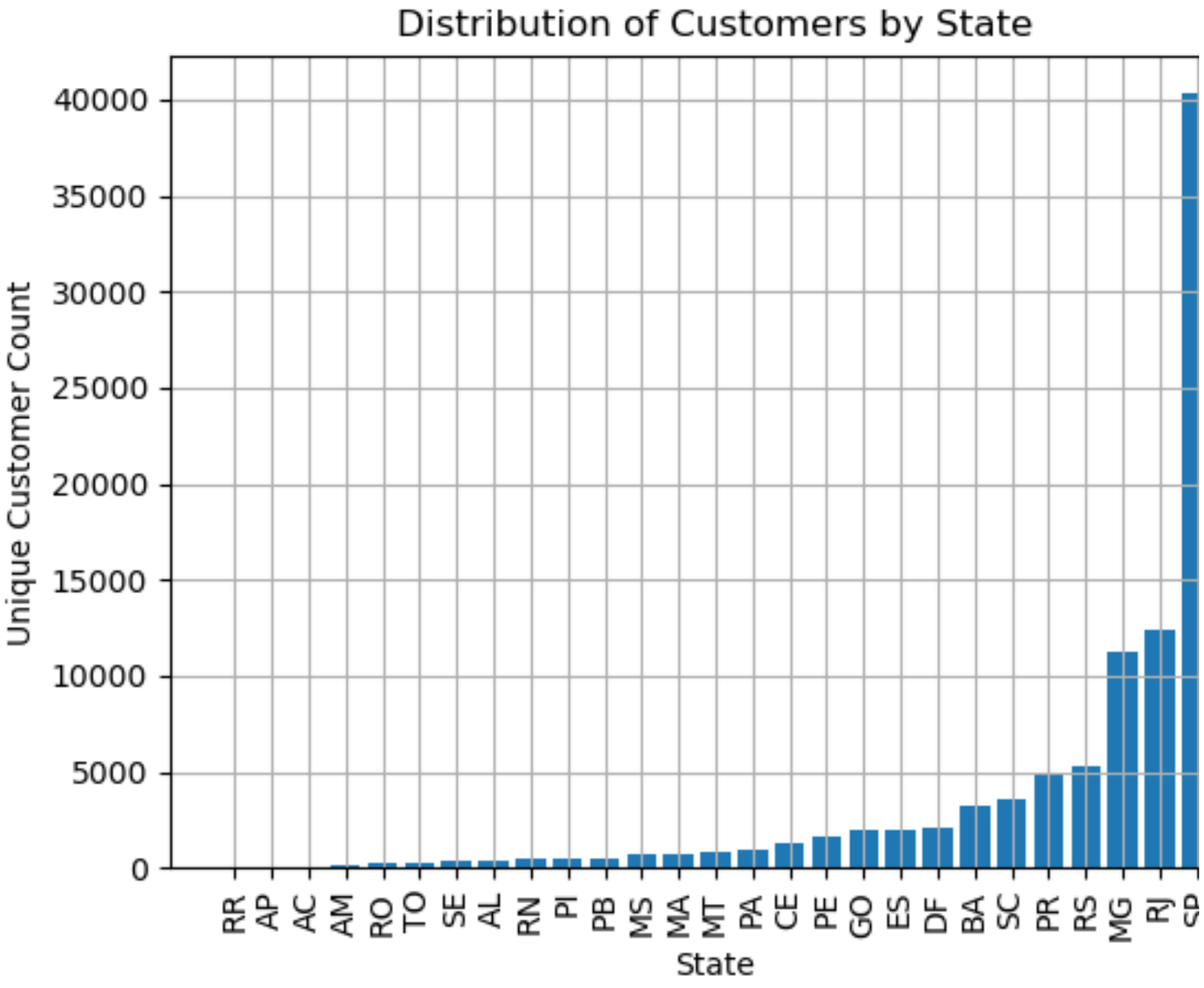
customer_unique_id - **L**'identifiant client unique.

customer_zip_code_prefix

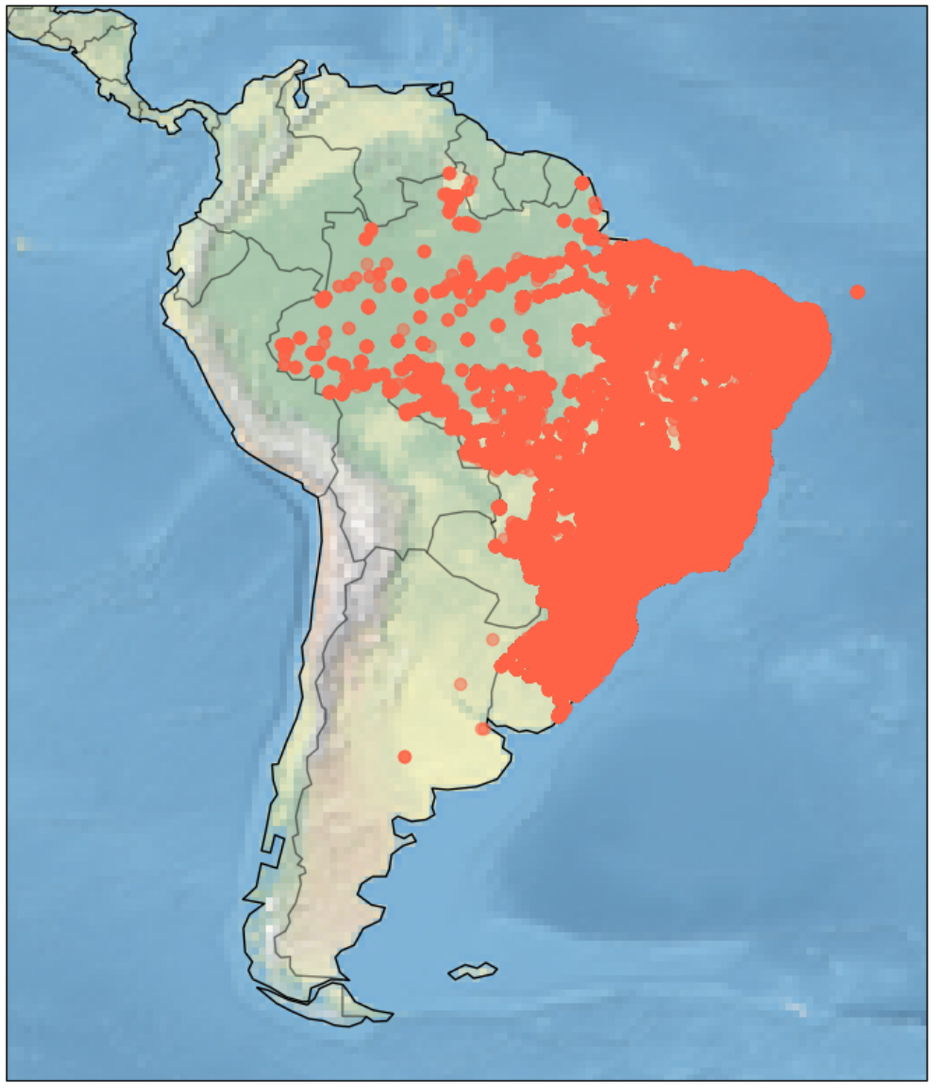
customer_city

customer_state

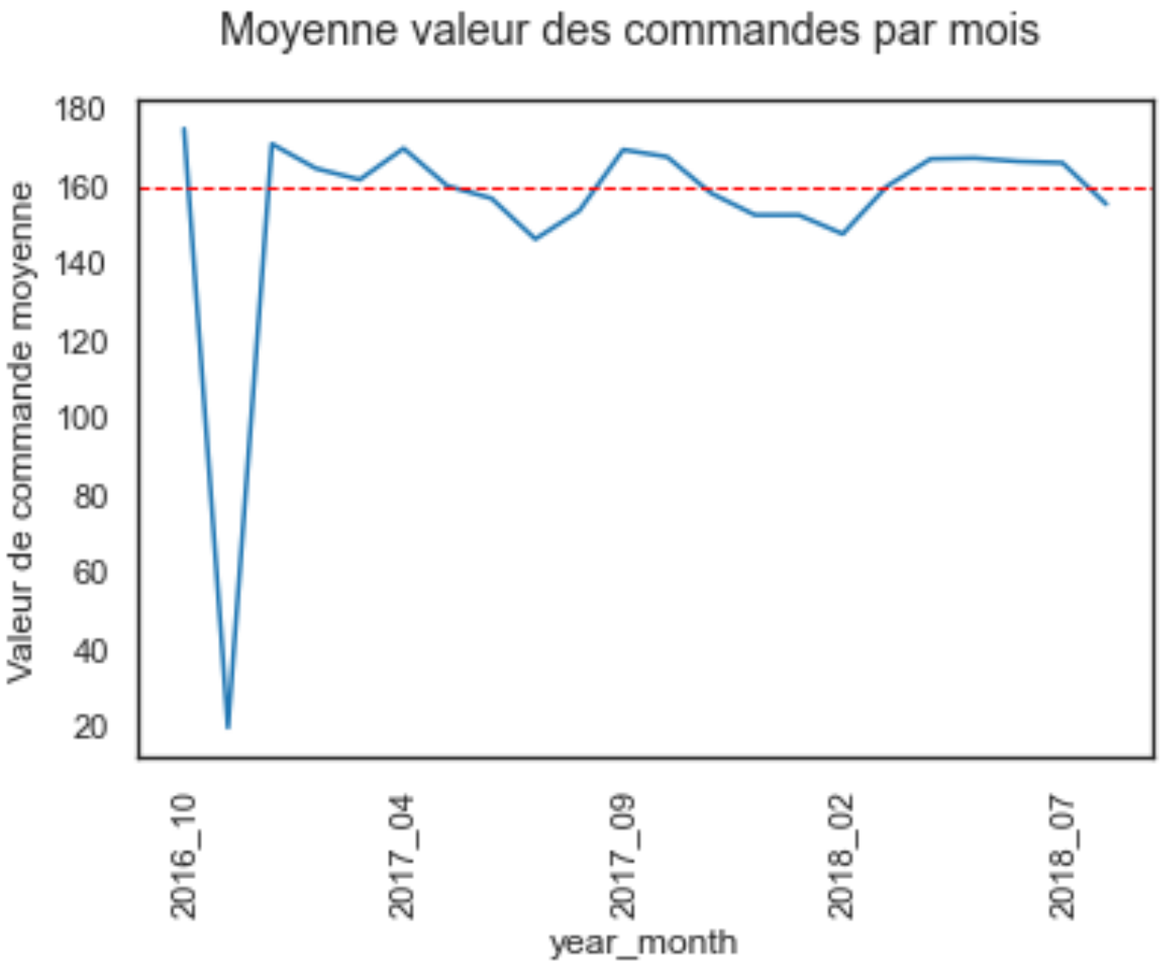
La plupart des clients vivent dans la capitale São Paulo et dans les villes voisines.
Cela est dû aux difficultés de logistique, au coût de livraison et à la répartition de la population.



La carte montre l'emplacement des commandes de produits achetés et, selon la répartition de la population, explique le plus haut niveau d'achats sur la côte Est. Les clients vivant dans le nord et le nord-est du Brésil doivent supporter des frais d'expédition plus élevés et attendre plus longtemps pour recevoir leur achat.



Analyse financier

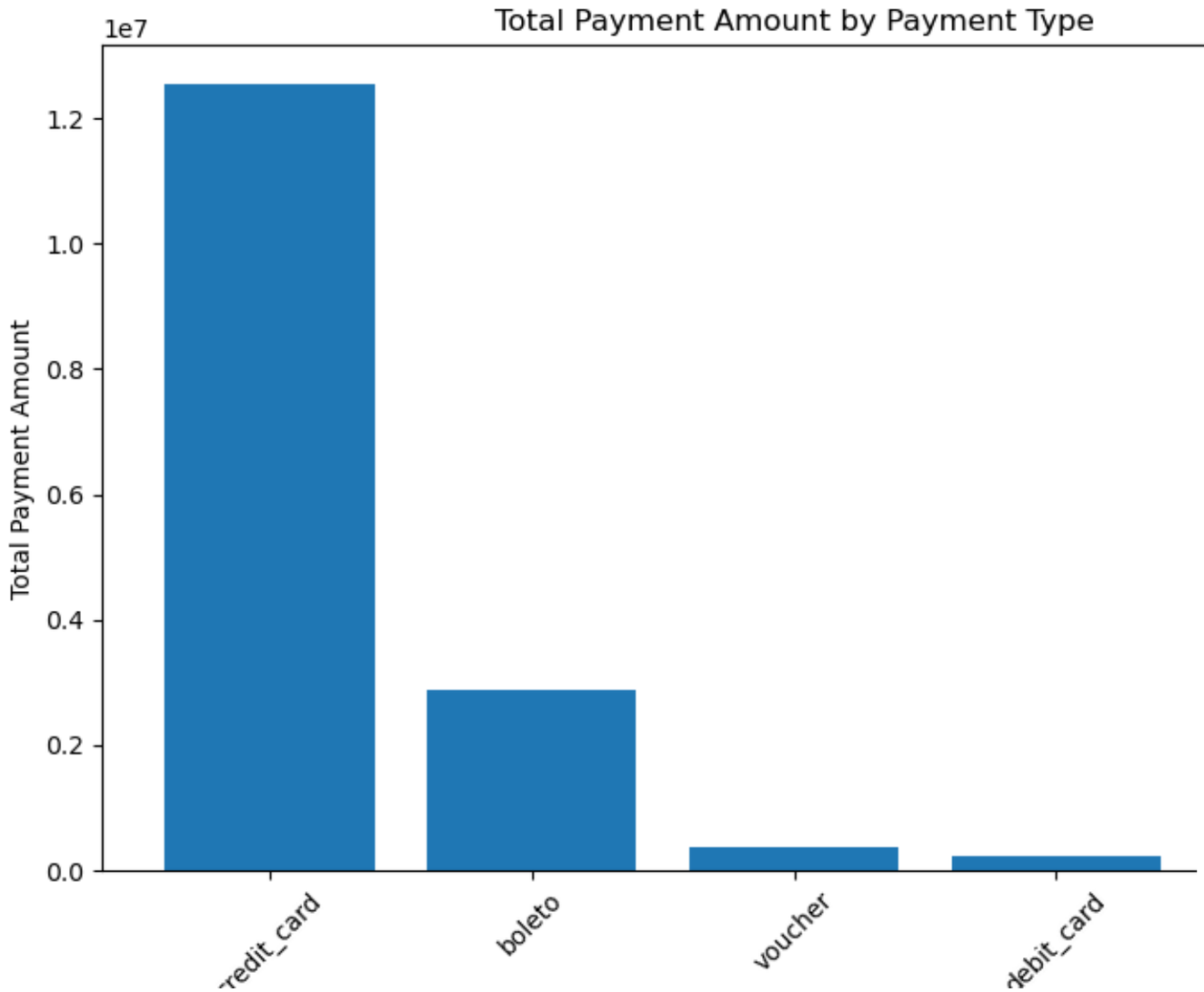


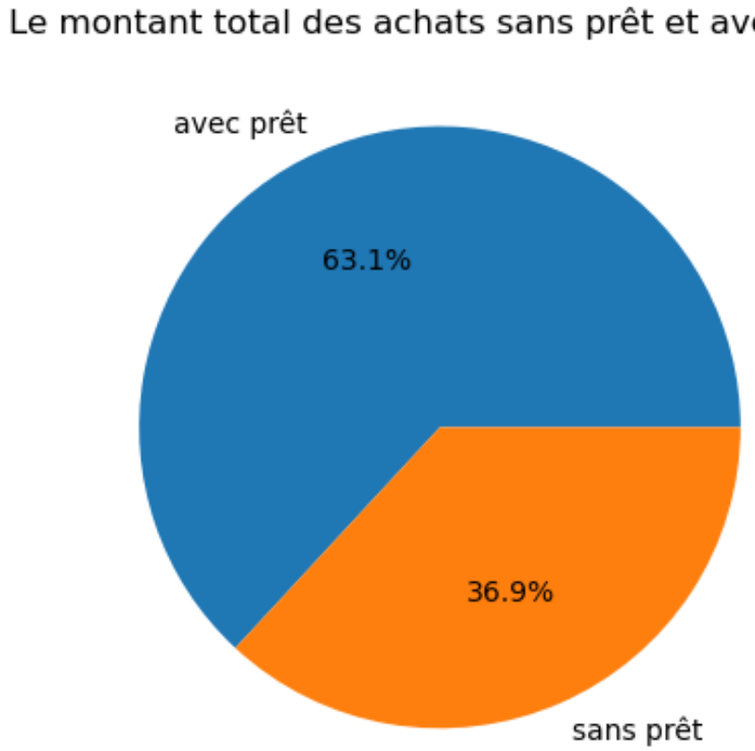
- Il y a des fluctuations saisonnières dans l'activité d'achat des clients (Black Friday, Noël, Nouvel An, Pâques).
- 97% des clients ont fait seulement un commande.
- Nombre de commandes - croissant pendant 1 an, puis stable.

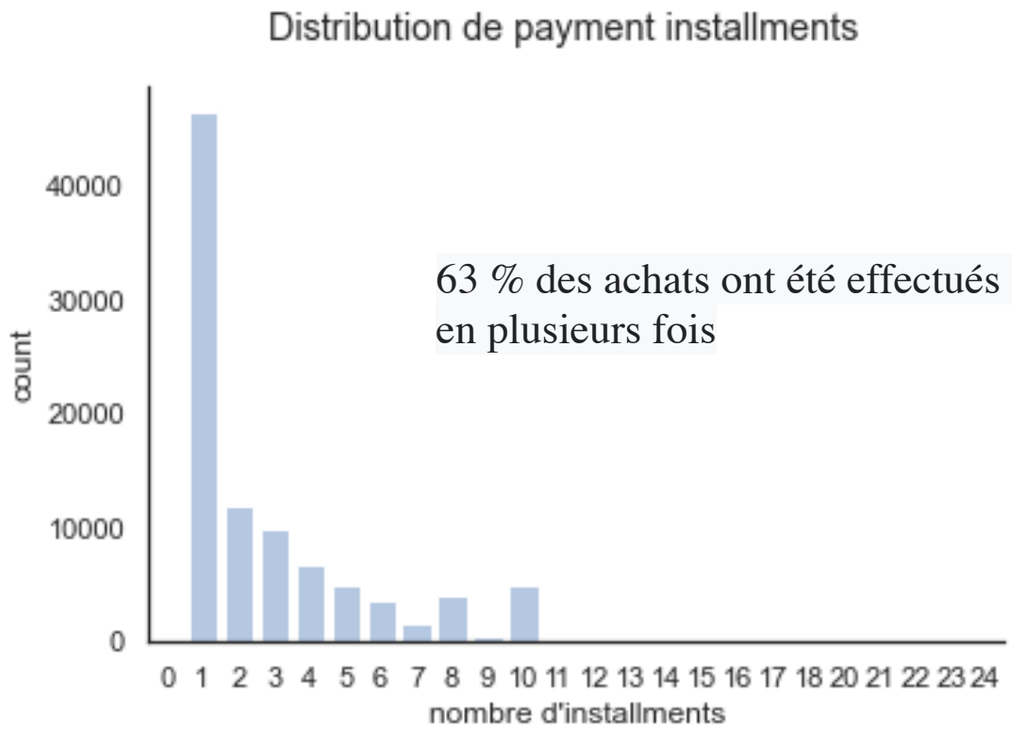
Analyse financier: moyen et nombre de paiements

payment_type	payment_value
credit_card	12542084.19
boleto	2869361.27
voucher	379436.87
debit_card	217989.79
not_defined	0.00

payment_type	count
credit_card	76795
boleto	19784
voucher	5775
debit_card	1529
not_defined	3







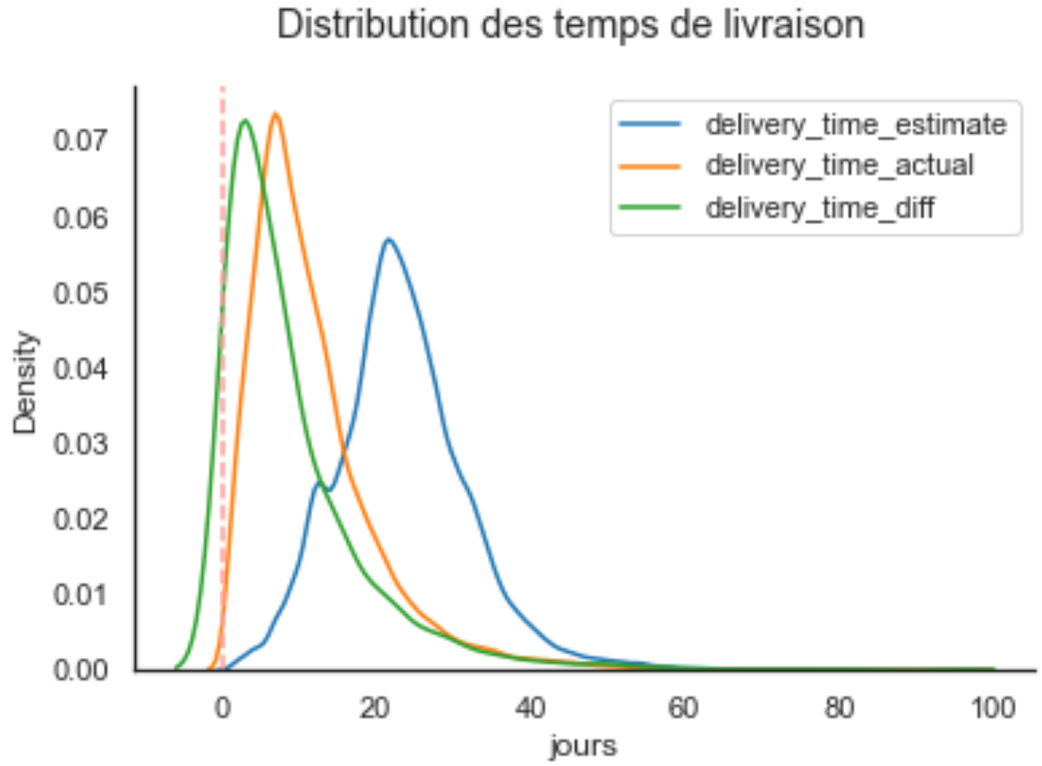
63 % des achats ont été effectués en plusieurs fois

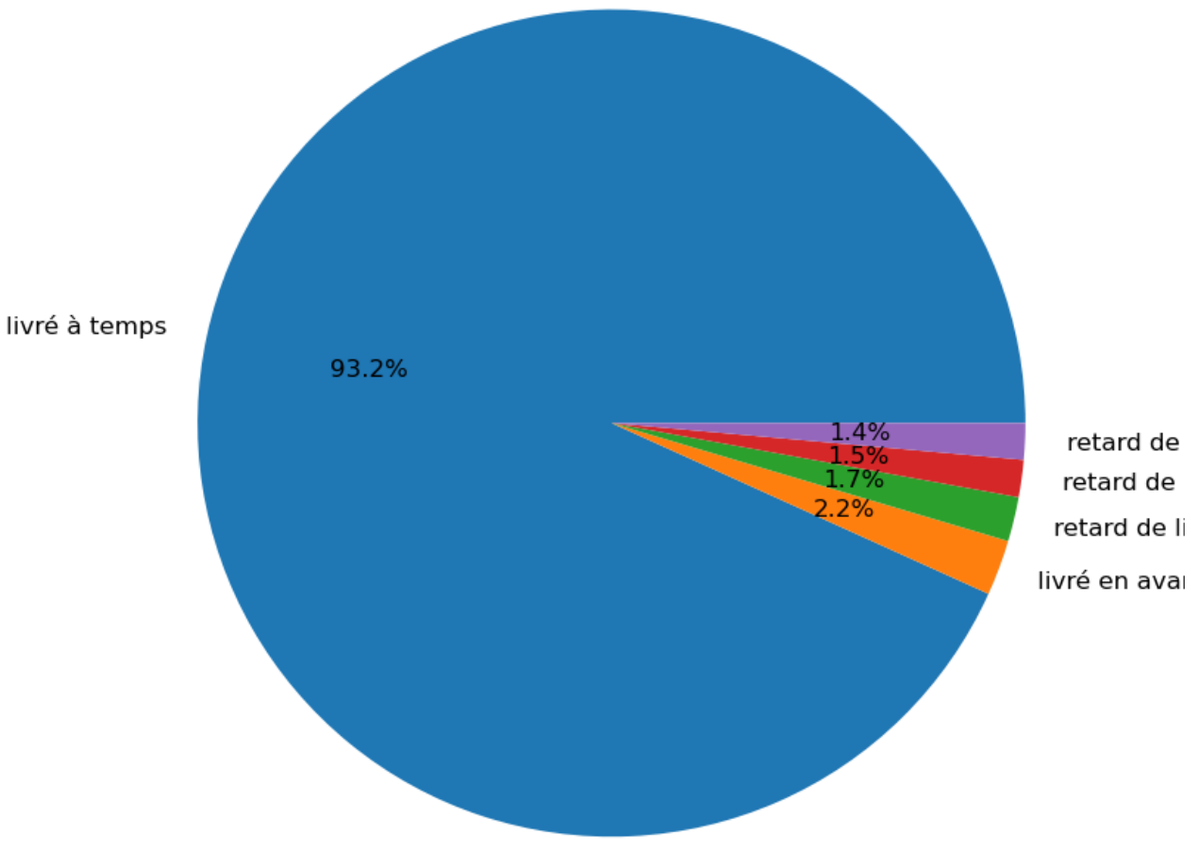
Analyse accessibilité : Délai de livraison

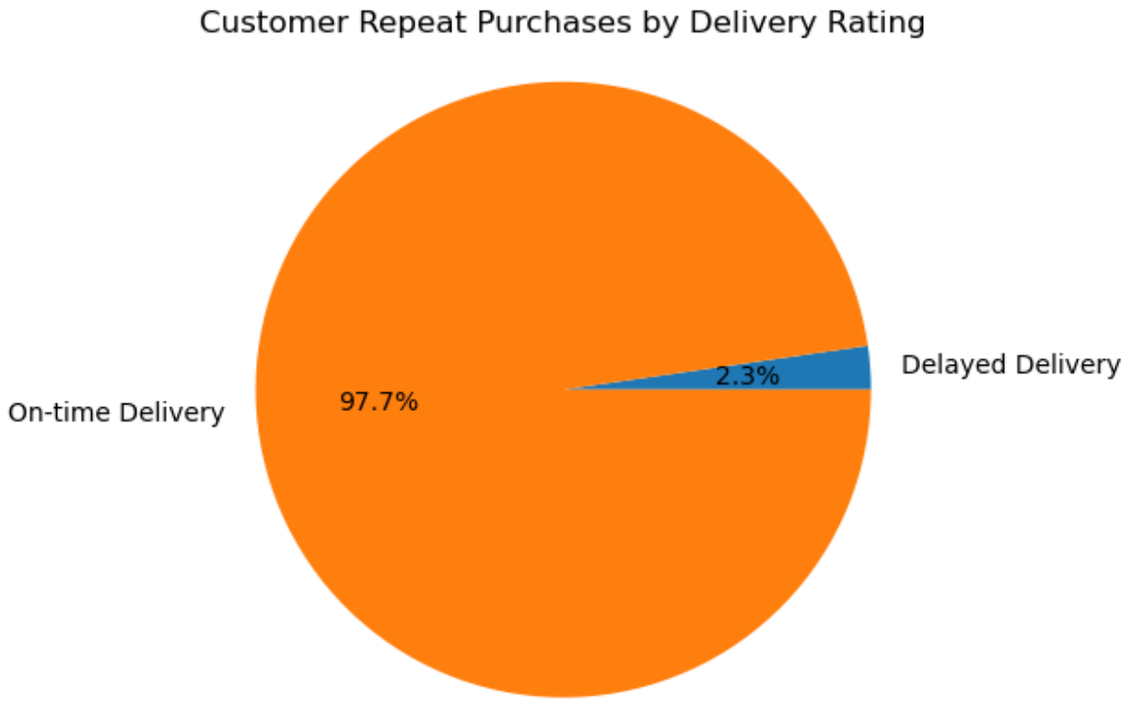
Voici le nombre de clients dans chaque catégorie de livraison :

Livré à temps : 89927 clients
Retard de livraison important : 2093 clients
Retard de livraison moyen : 1671 clients
Retard de livraison mineur : 1400 clients
Livré en avance : 1370 clients

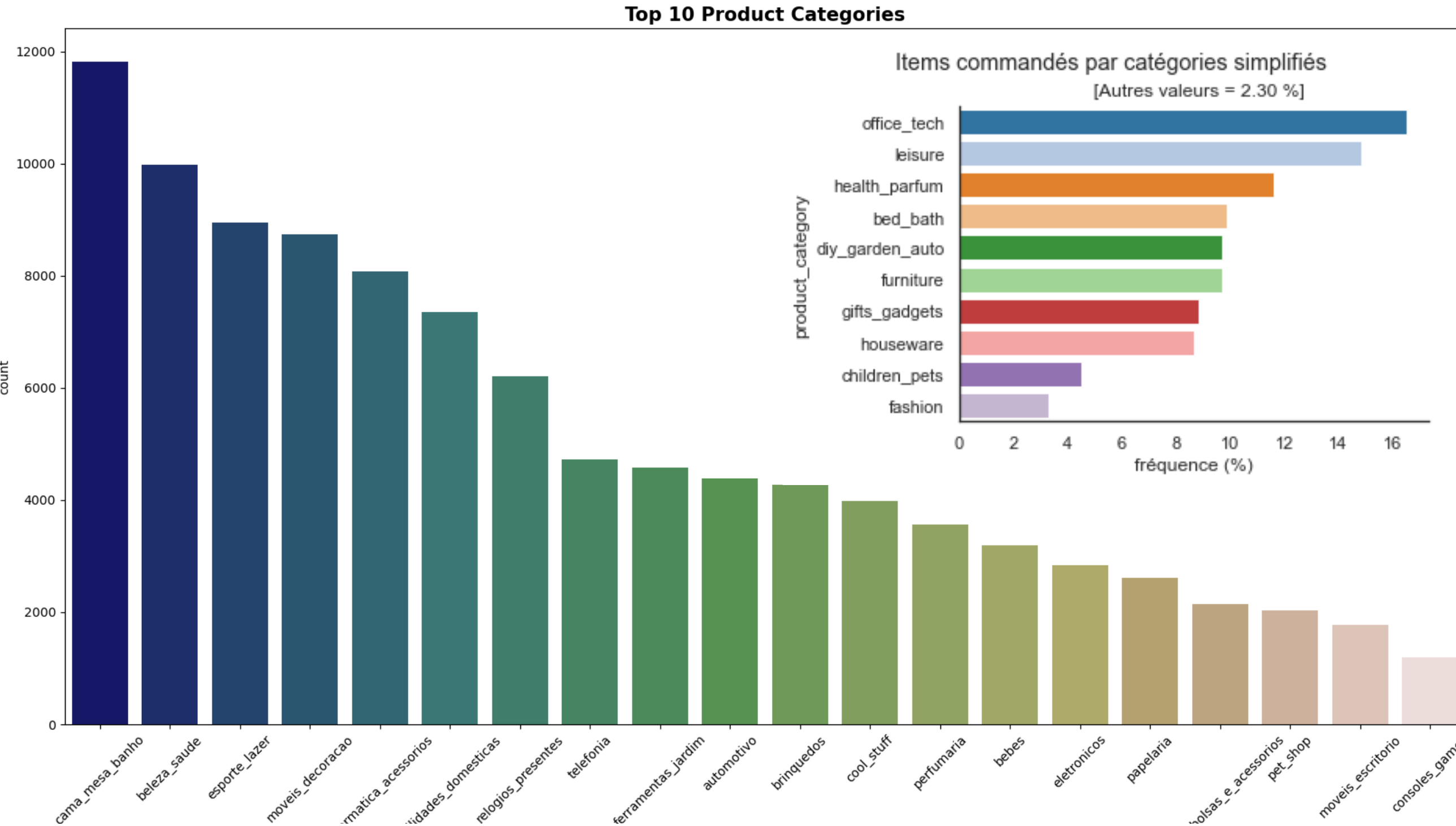
order_status	
delivered	96478
shipped	1107
canceled	625
unavailable	609
invoiced	314
processing	301
created	5
approved	2





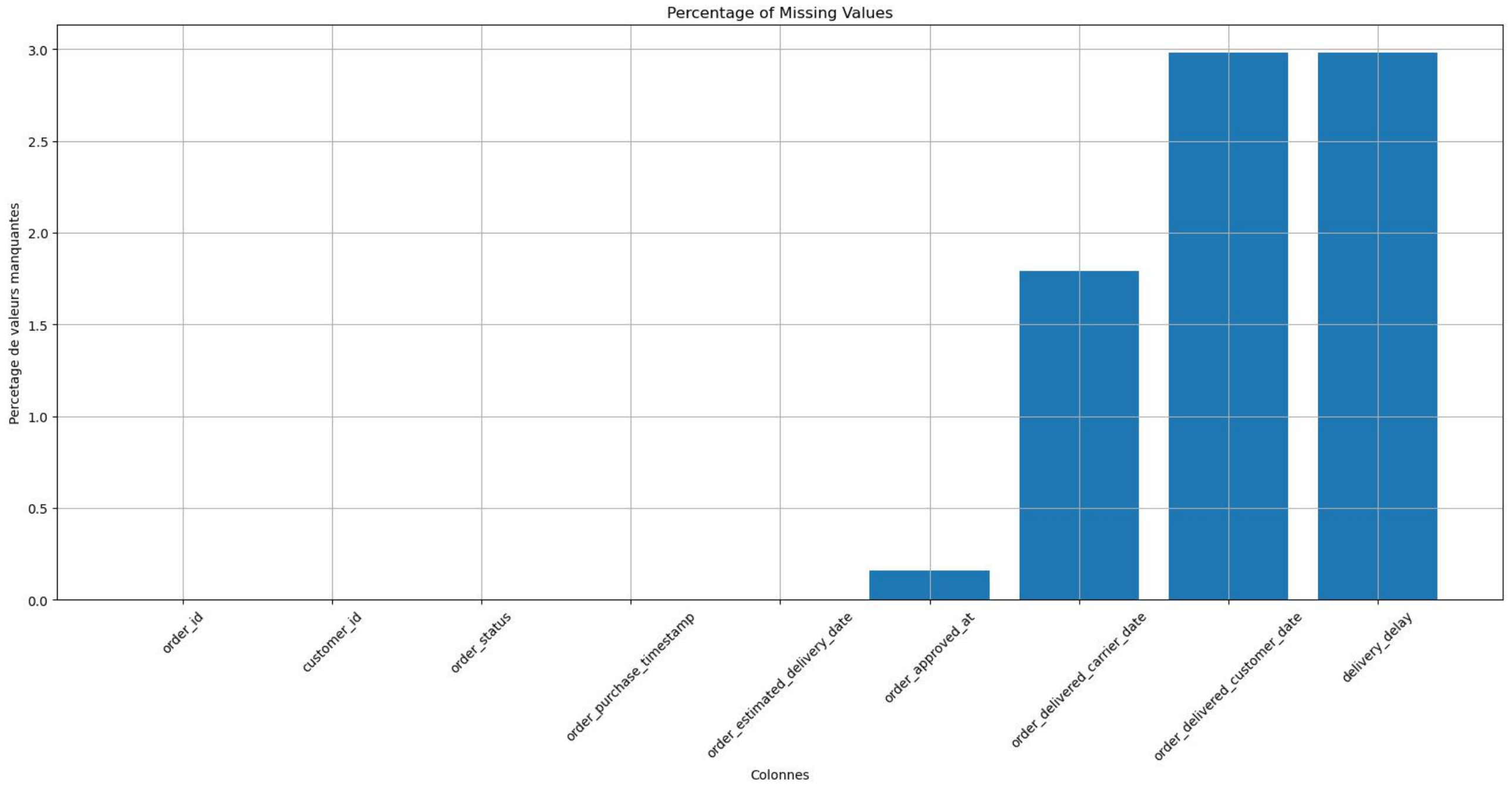


Analyse produits



Missing Values

Certaines valeurs liées aux dates de livraison et de réception des achats sont manquantes. En effet, certaines commandes ont été annulées ou incomplètes.



RFM (Récence, Fréquence, Monétaire)

En matière de marketing, si vous essayez de parler à tout le monde, vous aurez du mal à atteindre qui que ce soit.

Récence (R) - temps en jours entre le dernier achat et la date d'analyse. *Les clients qui ont récemment effectué un achat sont considérés comme plus susceptibles que ceux qui ont effectué un achat il y a longtemps.*

Fréquence (F) - le nombre d'achats effectués par le client. *Des achats fréquents peuvent indiquer une grande fidélité des clients.*

Monétaire (M) - le montant total que le client a dépensé en achats. *Les clients qui ont dépensé plus d'argent peuvent être plus précieux pour une entreprise.*

```
frequency = data.groupby(`customer_unique_id')['order_id'].nunique()

df['total_value'] = df['price'] + df[`freight_value']
monetary = data.groupby('customer_unique_id')['total_value'].sum()

last_purchase = data.groupby('customer_unique_id')['order_purchase_timestamp'].max()
recency = (last_purchase.max() - last_purchase).dt.days

rfm_data = pd.DataFrame({'recency': recency, 'frequency': frequency, 'monetary': monetary})
```

	recency	frequency	monetary
count	93396.000000	93396.000000	93396.000000
mean	241.744250	1.033406	174.134824
std	153.164721	0.208235	263.116852
min	0.000000	1.000000	10.070000
25%	118.000000	1.000000	64.000000
50%	222.000000	1.000000	110.440000
75%	351.000000	1.000000	189.000000
max	728.000000	15.000000	13664.080000

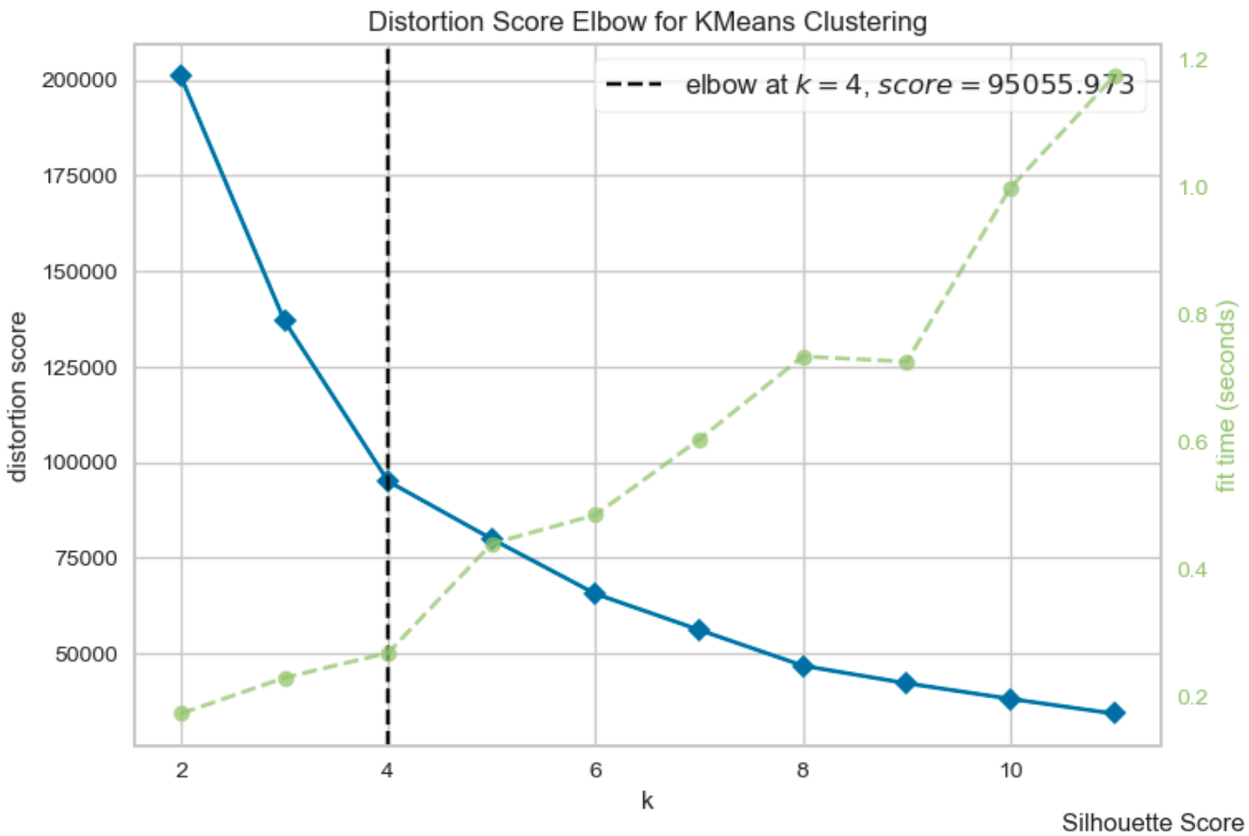
rfm_data

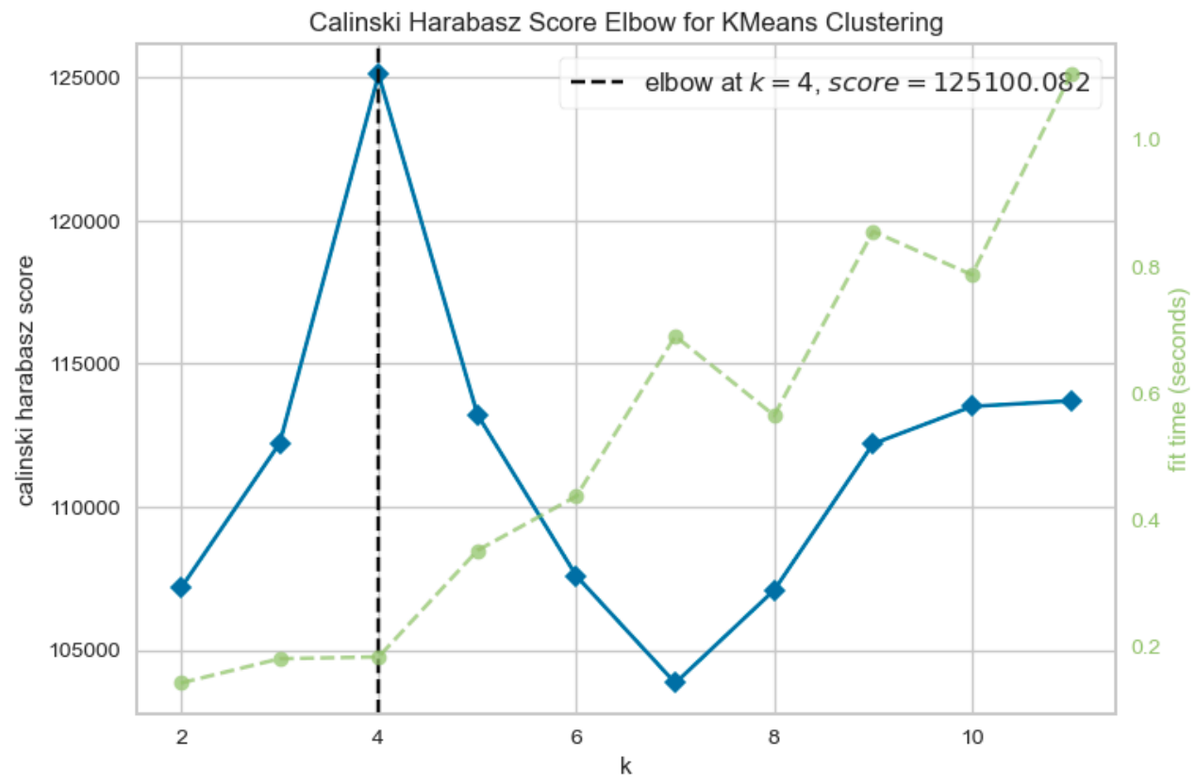
customer_unique_id	Recency	Frequency	Monetary
0000366f3b9a7992bf8c76cfdf3221e2	115	1	141.90
0000b849f77a49e4a4ce2b2a4ca5be3f	118	1	27.19
0000f46a3911fa3c0805444483337064	541	1	86.22
0000f6ccb0745a6a4b88665a16c9f078	325	1	43.62
0004aac84e0df4da2b147fca70cf8255	292	1	196.89
...	...	...	...
fffcf5a5ff07b0908bd4e2dbc735a684	451	1	2067.42
fffea47cd6d3cc0a88bd621562a9d061	266	1	84.58
ffff371b4d645b6ecea244b27531430a	572	1	112.46
ffff5962728ec6157033ef9805bacc48	123	1	133.69
ffffd2657e2aad2907e67c3e9daecbeb	488	1	71.56

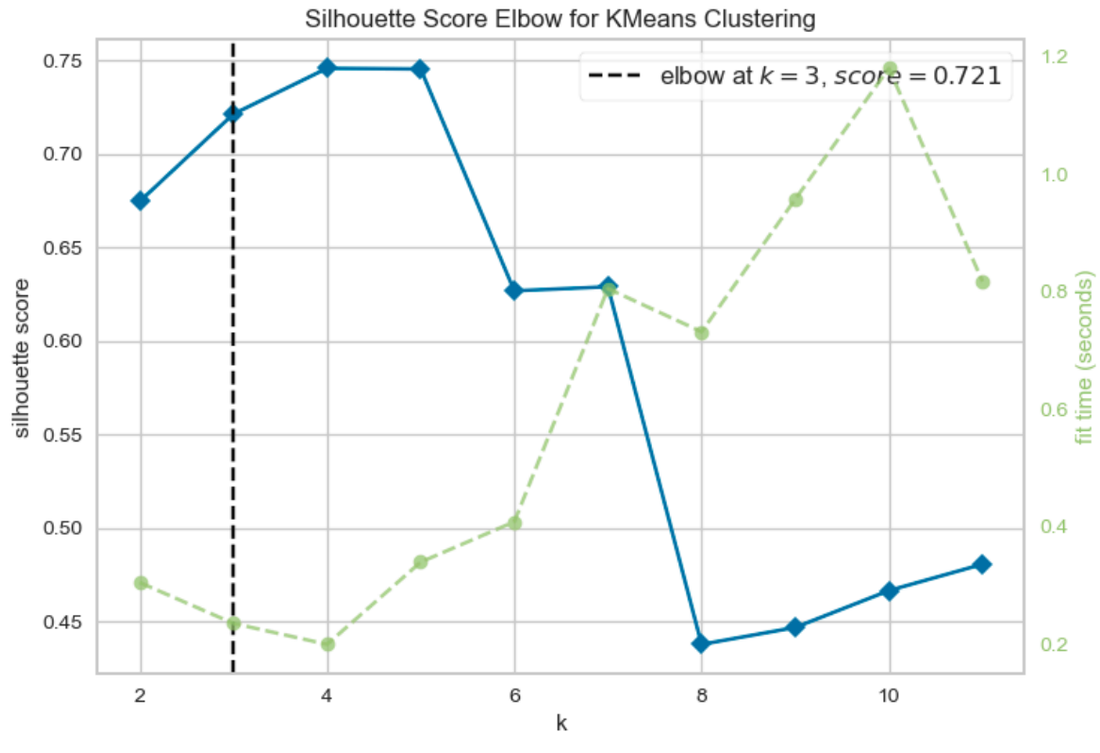
93396 rows × 3 columns

Segmentation **Kmeans** (Récence, Fréquence, Monétaire)

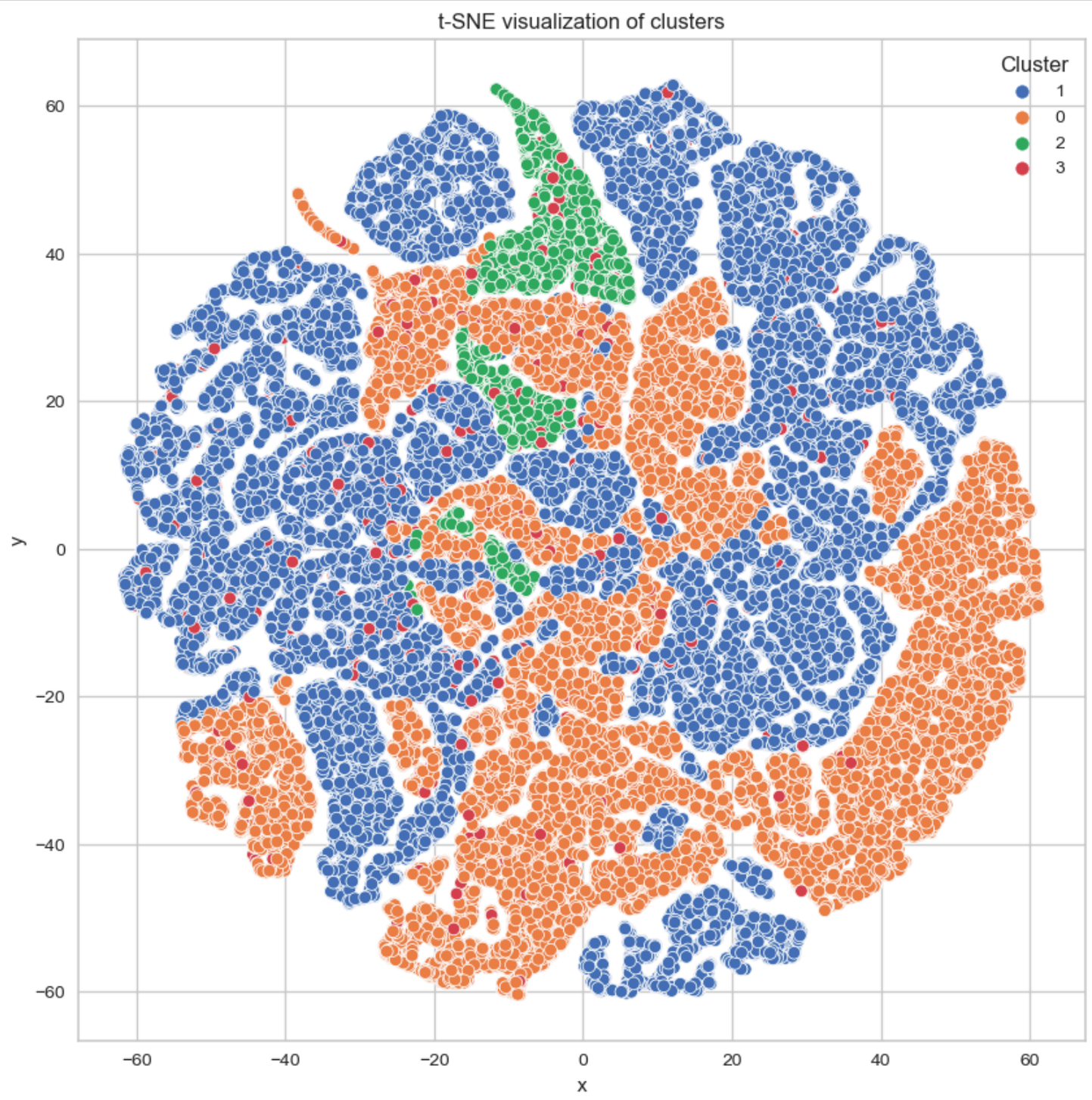
Choix du nombre de clusters (k)

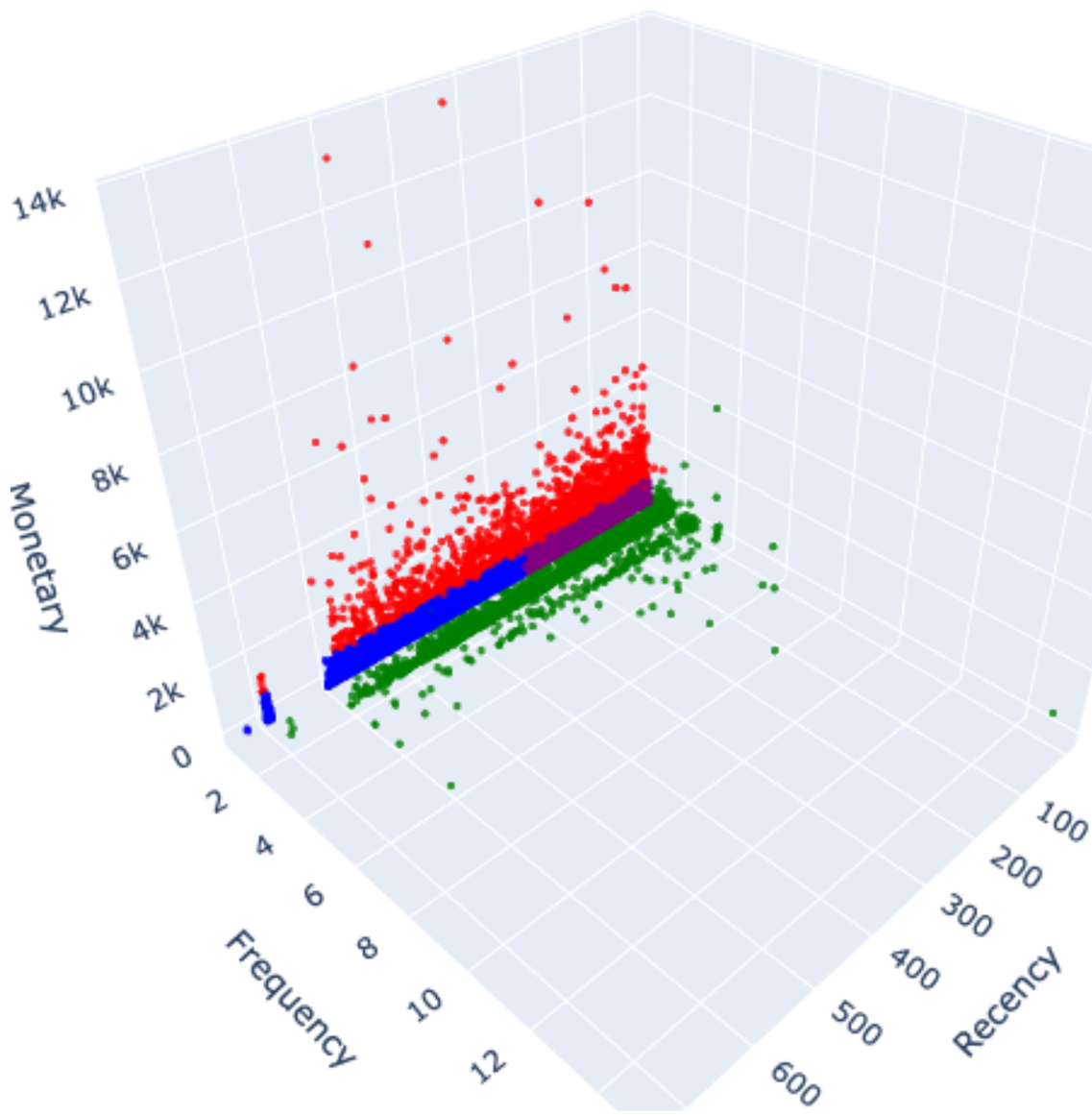


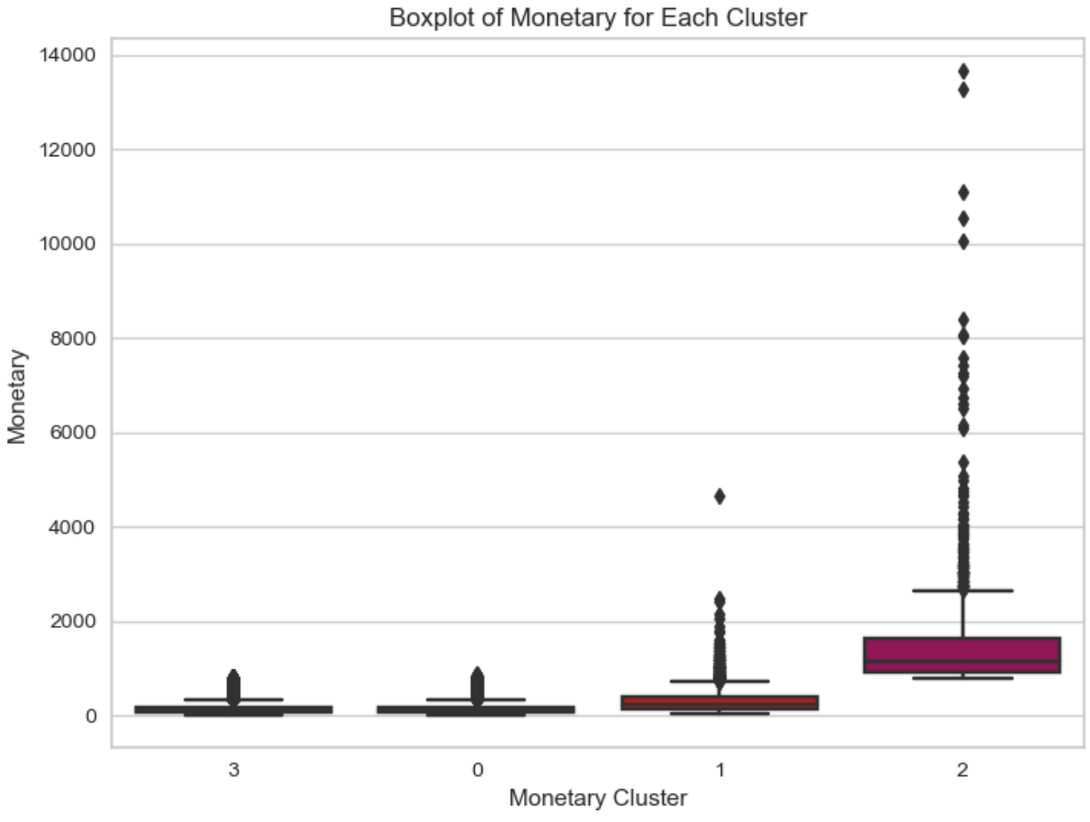


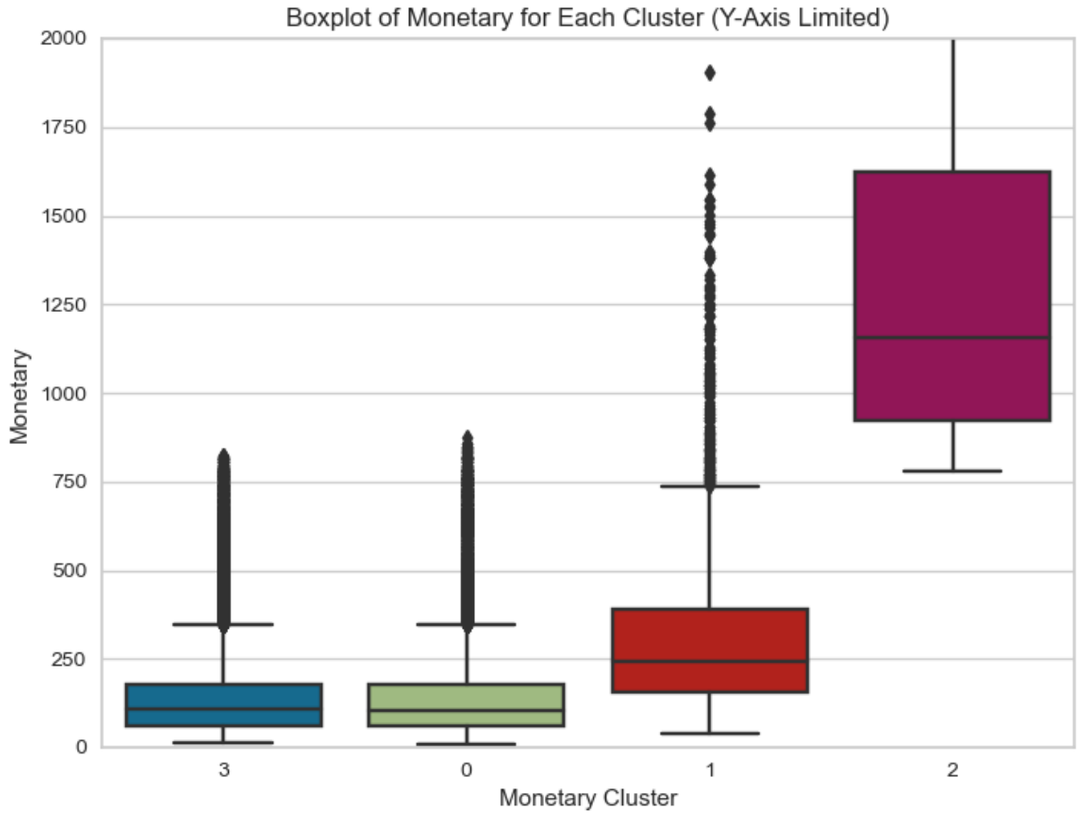


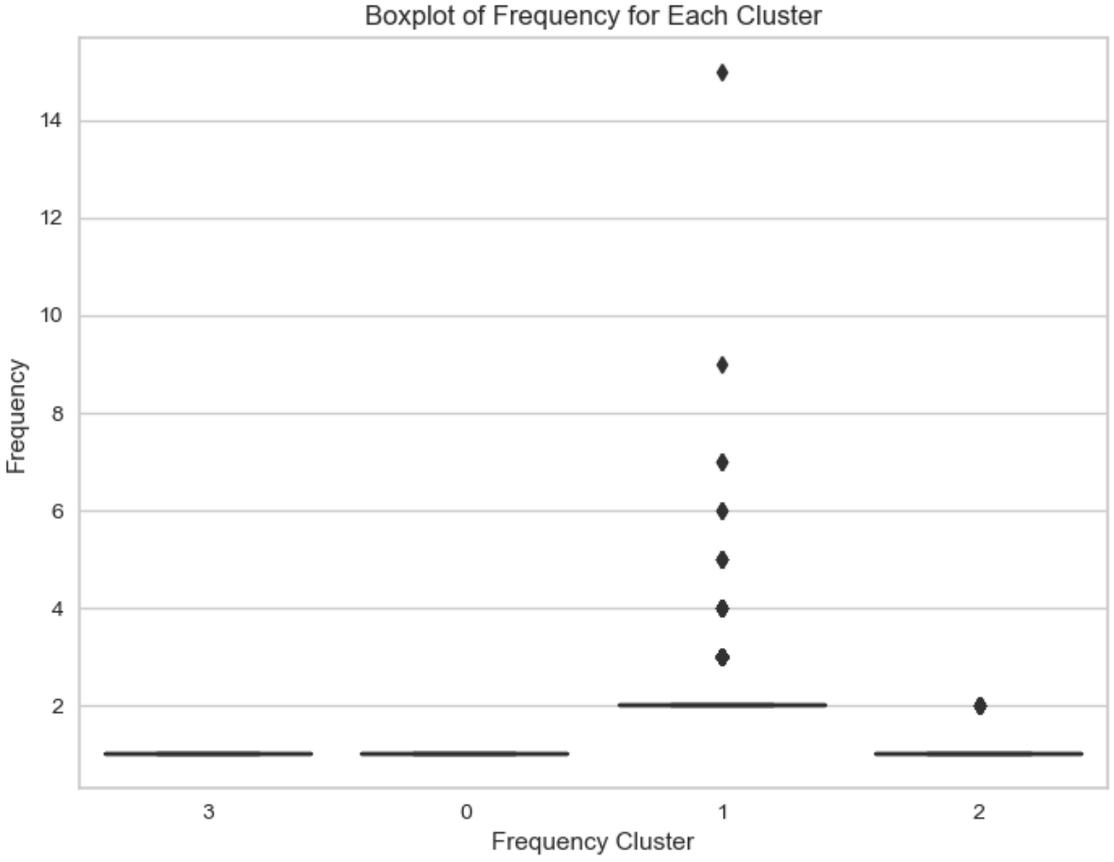
Segmentation par Kmeans (K=4)

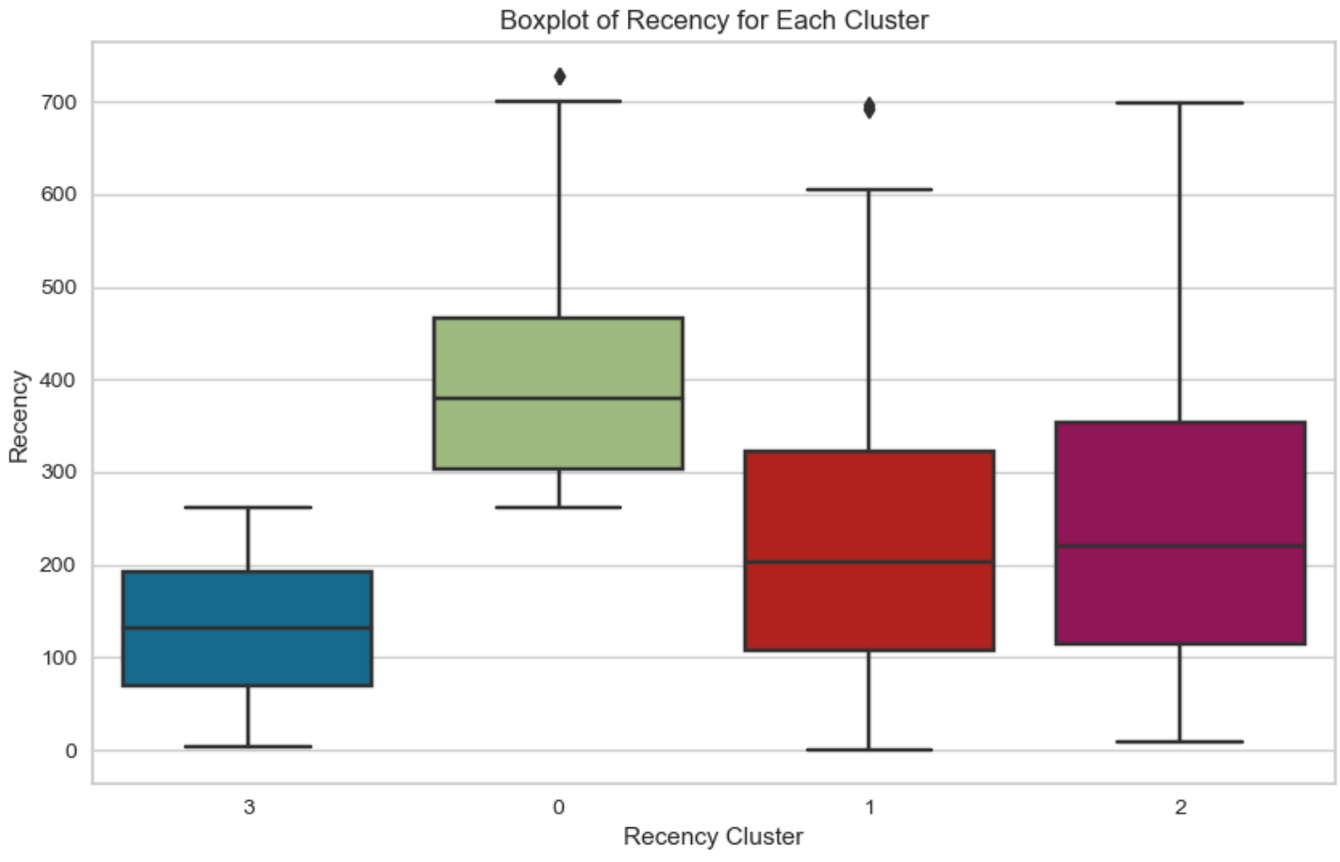




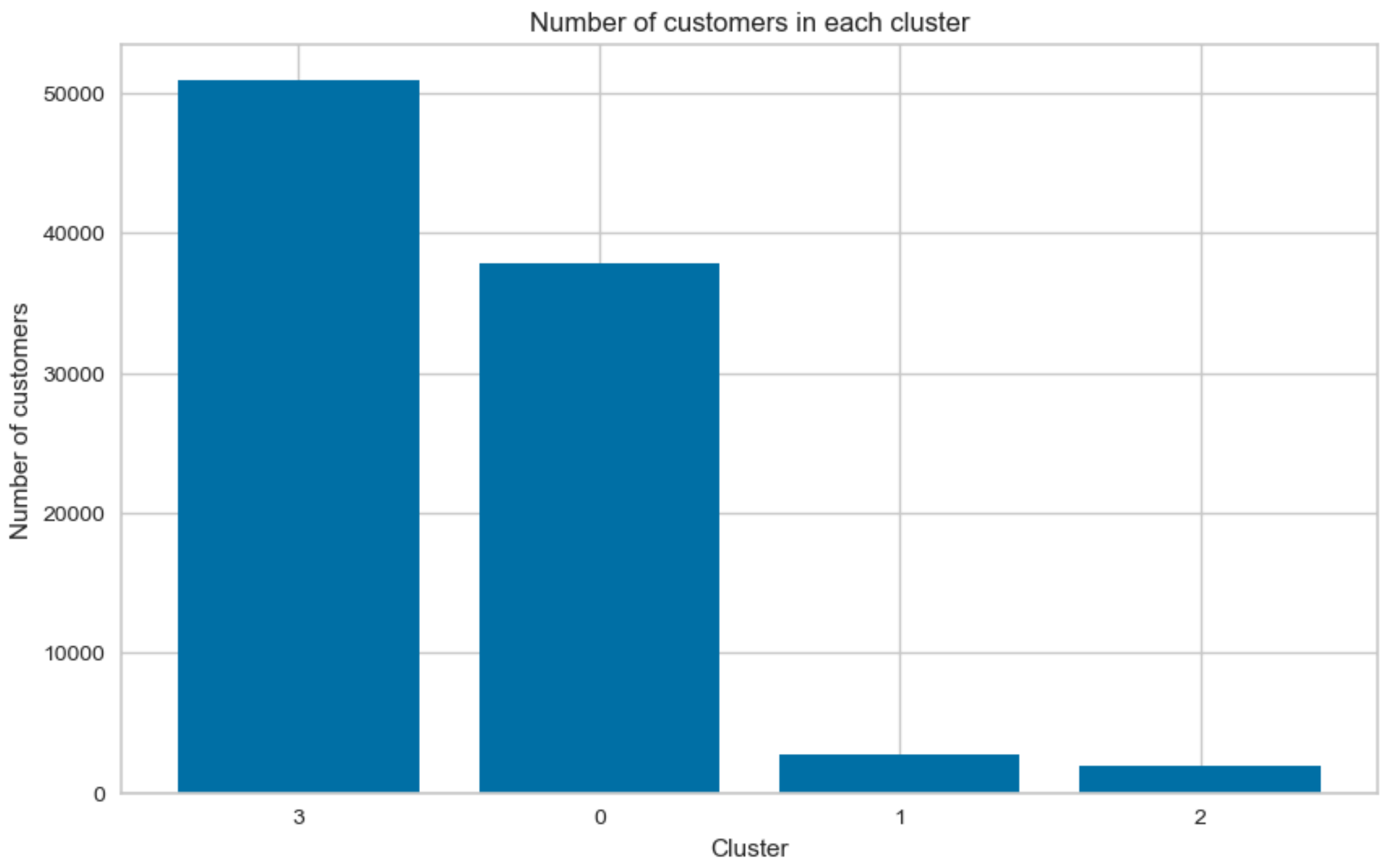

Boxplot of Monetary for Each Cluster
Monetary
Monetary Cluster
14000
12000
10000
8000
6000
4000
2000
0
3
0
1
2

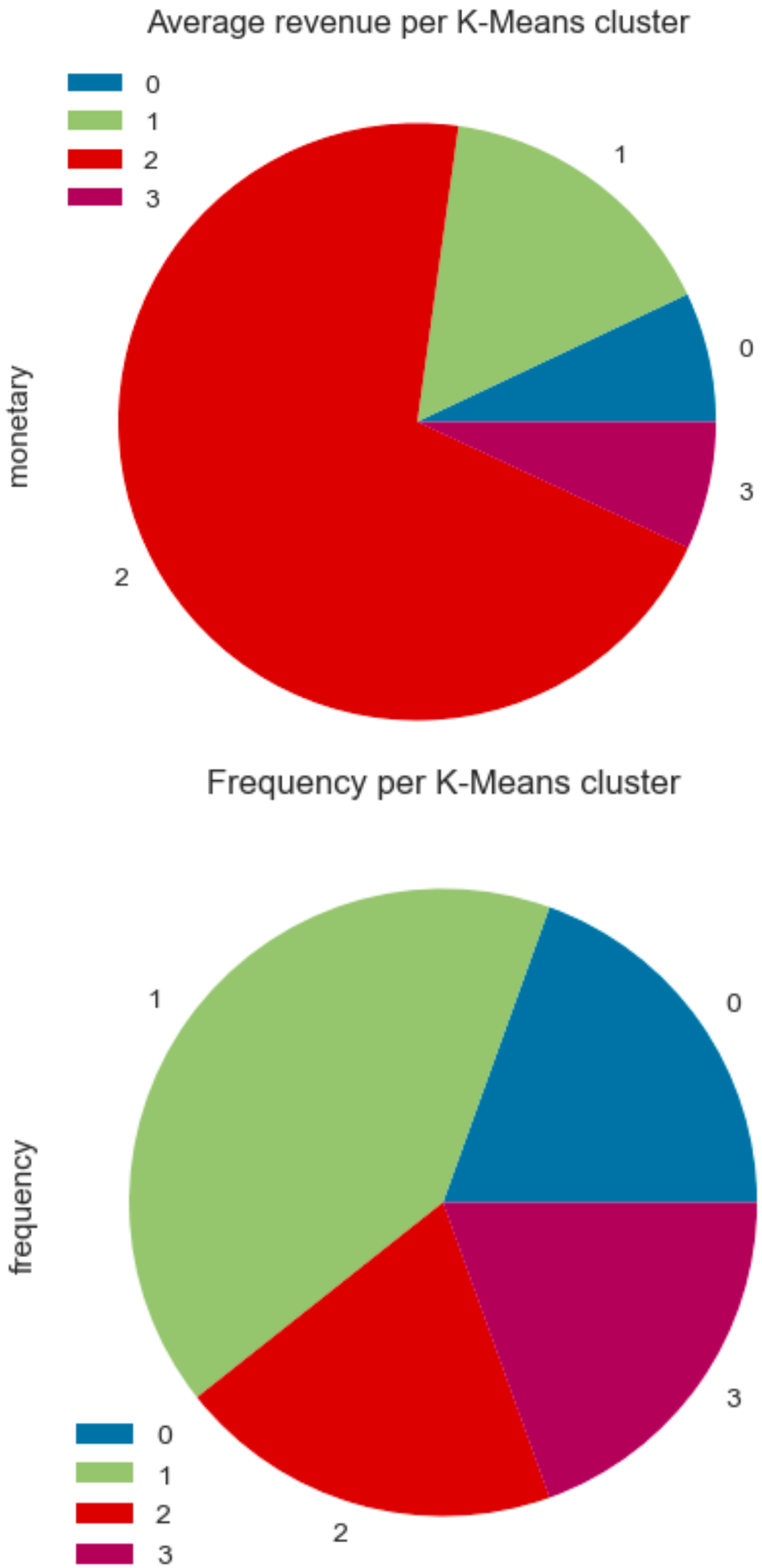

Boxplot of Monetary for Each Cluster (Y-Axis Limited)
Monetary
Monetary Cluster
2000
1750
1500
1250
1000
750
500
250
0
3
0
1
2

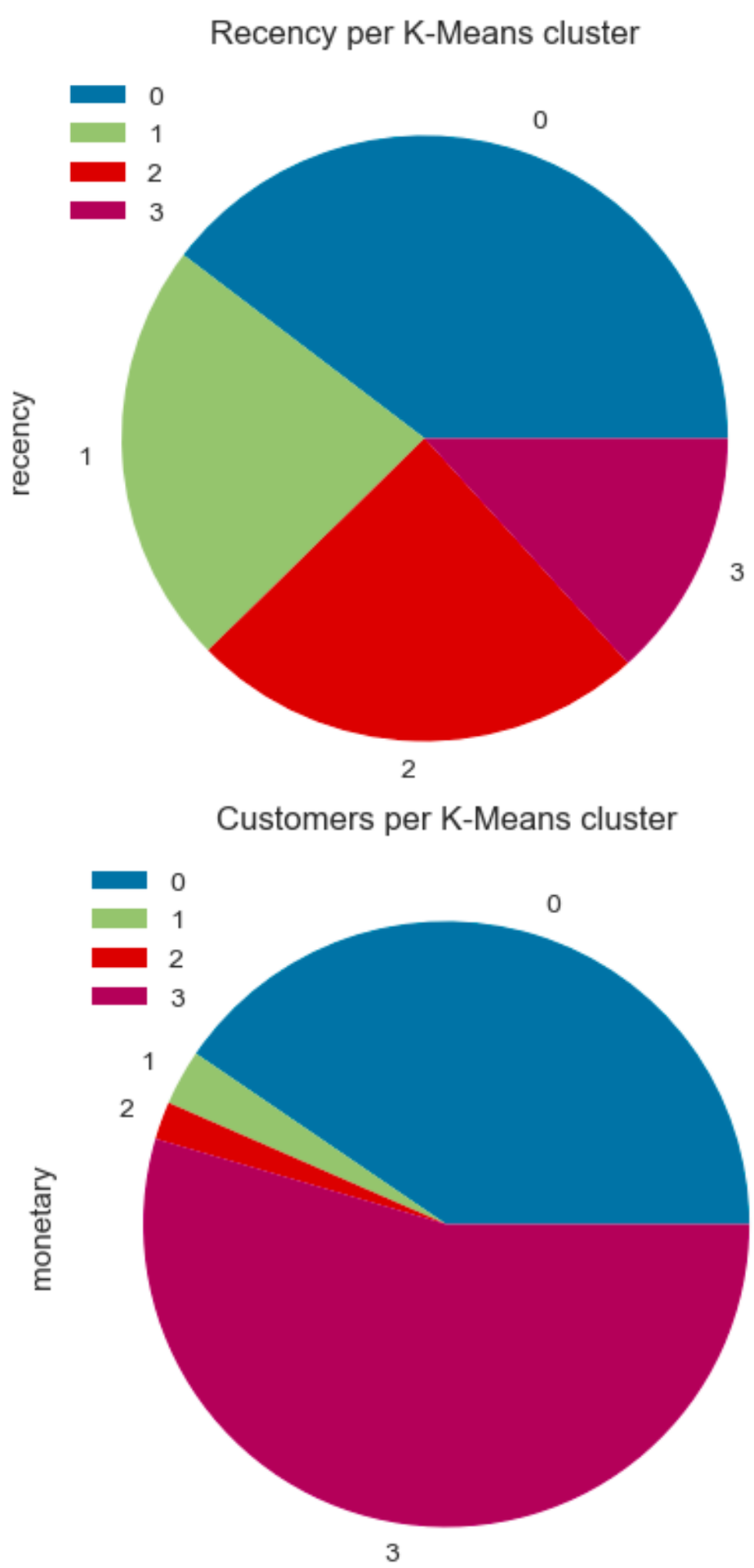

Boxplot of Frequency for Each Cluster
Frequency
Frequency Cluster
14
12
10
8
6
4
2
3
0
1
2


Boxplot of Recency for Each Cluster
Recency
Recency Cluster
700
600
500
400
300
200
100
0
3
0
1
2

	recency					frequency					monetary					
	std	mean	median	min	max	std	mean	median	min	max	std	mean	median	min	max	count
Cluster																
0	96.0	386.0	374.0	256	694	0.0	1.0	1.0	1	1	95.0	130.0	102.0	10.0	677.0	35964
1	72.0	126.0	127.0	0	256	0.0	1.0	1.0	1	1	91.0	128.0	103.0	11.0	539.0	48616
2	145.0	232.0	216.0	0	693	0.0	1.0	1.0	1	2	729.0	966.0	742.0	437.0	13664.0	4213
3	144.0	218.0	197.0	0	691	0.0	2.0	2.0	2	14	287.0	329.0	245.0	37.0	4656.0	2688




Average revenue per K-Means cluster
0
1
2
3
monetary
Frequency per K-Means cluster
frequency


Recency per K-Means cluster
0
1
2
3
recency
Customers per K-Means cluster
monetary

La description de la segmentation de la clientèle

 # Cluster 1 - *Clients récents*

Ces clients ont récemment effectué leur achat, il peut donc être efficace de les cibler avec des campagnes promotionnelles pour de nouveaux produits ou offres, des campagnes marketing personnalisées pour augmenter leur rétention et les convertir en nos meilleurs clients.

 # Cluster 2 - *VIP-clients*

Ces clients dépensent plus d'argent sur les achats.
Ils doivent proposer des offres plus luxueuses, des niveaux d'abonnement plus élevés et des ventes croisées/incitatives qui augmentent la valeur moyenne des commandes. Là encore, il n'est peut-être pas logique d'augmenter les marges en offrant des remises.

 # Cluster 3 - *Clients réguliers*

Ces clients achètent plus souvent.
Ils répondent mieux aux recommandations de produits basées sur les achats passés, ainsi qu'aux incitations à seuil (comme un cadeau gratuit pour les transactions supérieures à la valeur moyenne des commandes de la marque). Peut-être peuvent-ils être encouragés à acheter encore plus avec des offres spéciales pour les clients fidèles et à utiliser des campagnes de parrainage afin d'attirer de nouveaux clients.

 # Cluster 0 - *Groupe de risque*

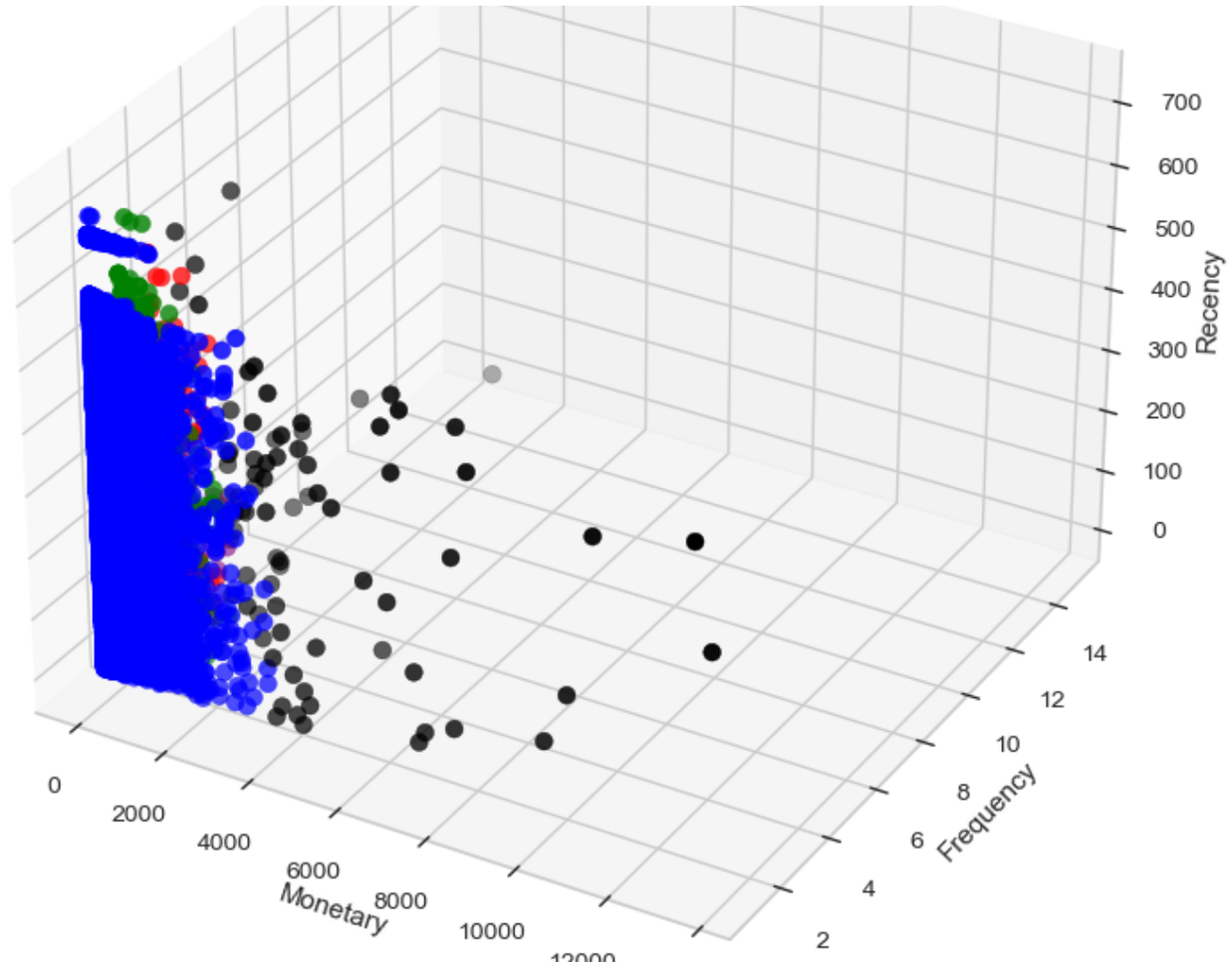
Étant donné que ces clients n'ont effectué qu'un seul achat il y a longtemps, une stratégie de récupération peut être utile. Par exemple, envoyez-leur des e-mails avec des rappels ou des offres spéciales pour les encourager à magasiner à nouveau.

Segmentation DBSCAN

	recency					frequency					monetary					
	std	mean	median	min	max	std	mean	median	min	max	std	mean	median	min	max	count
Cluster_dbscan																
-1	183.0	253.0	216.0	11	605	2.0	3.0	2.0	1	15	2527.0	2791.0	1902.0	47.0	13664.0	160
0	97.0	392.0	379.0	261	728	0.0	1.0	1.0	1	1	127.0	144.0	105.0	10.0	876.0	37796
1	150.0	239.0	218.0	8	698	0.0	1.0	1.0	1	1	503.0	1301.0	1130.0	780.0	3358.0	1791
2	144.0	226.0	206.0	0	696	0.0	2.0	2.0	2	2	223.0	299.0	233.0	37.0	1545.0	2543
3	140.0	198.0	163.0	5	572	0.0	3.0	3.0	3	3	221.0	402.0	359.0	88.0	1026.0	159
4	72.0	131.0	132.0	4	261	0.0	1.0	1.0	1	1	122.0	142.0	106.0	11.0	822.0	50947

```
Best Parameters: {'eps': 1.5, 'min_samples': 15}
Silhouette Score: 0.7196035649174892
```

Les résultats sont très sensibles au choix d'hyperparamètres



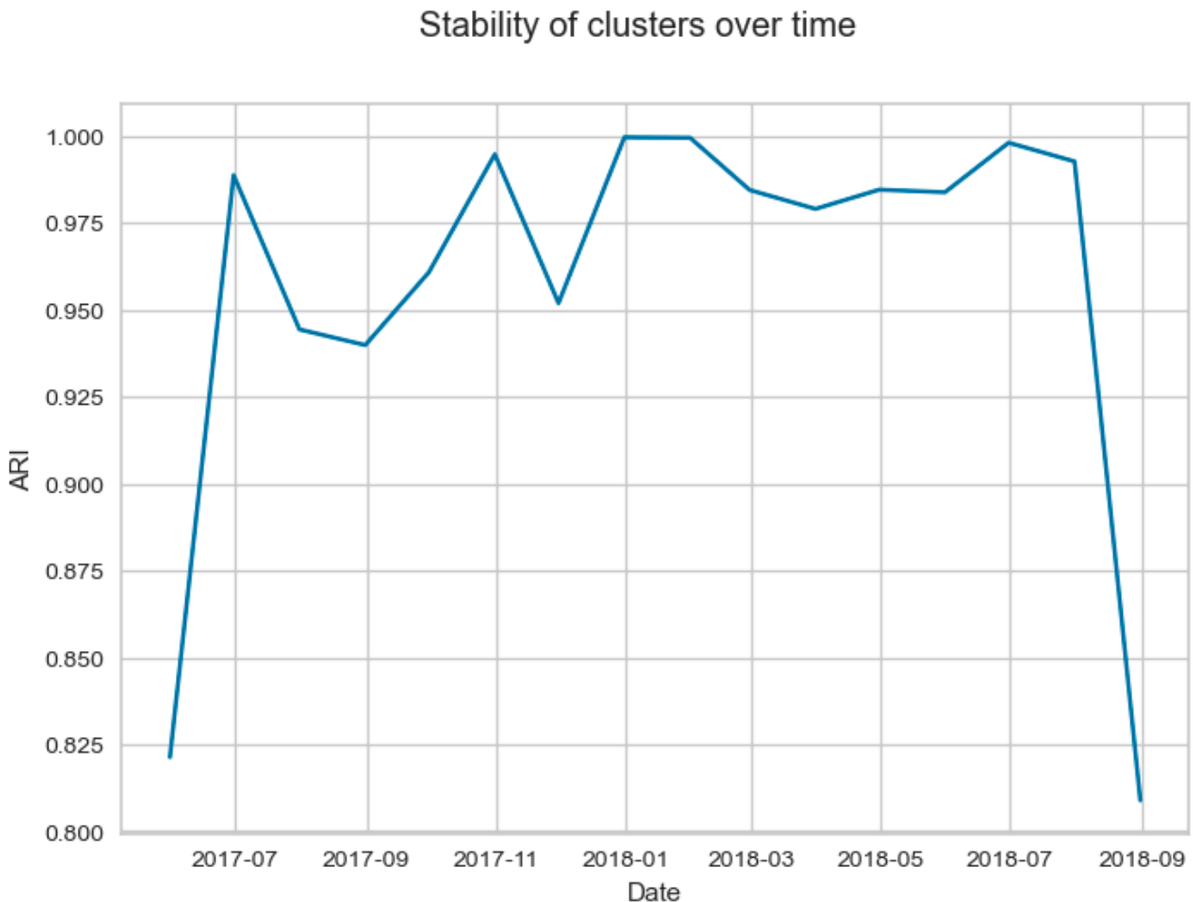
Stabilité des segments

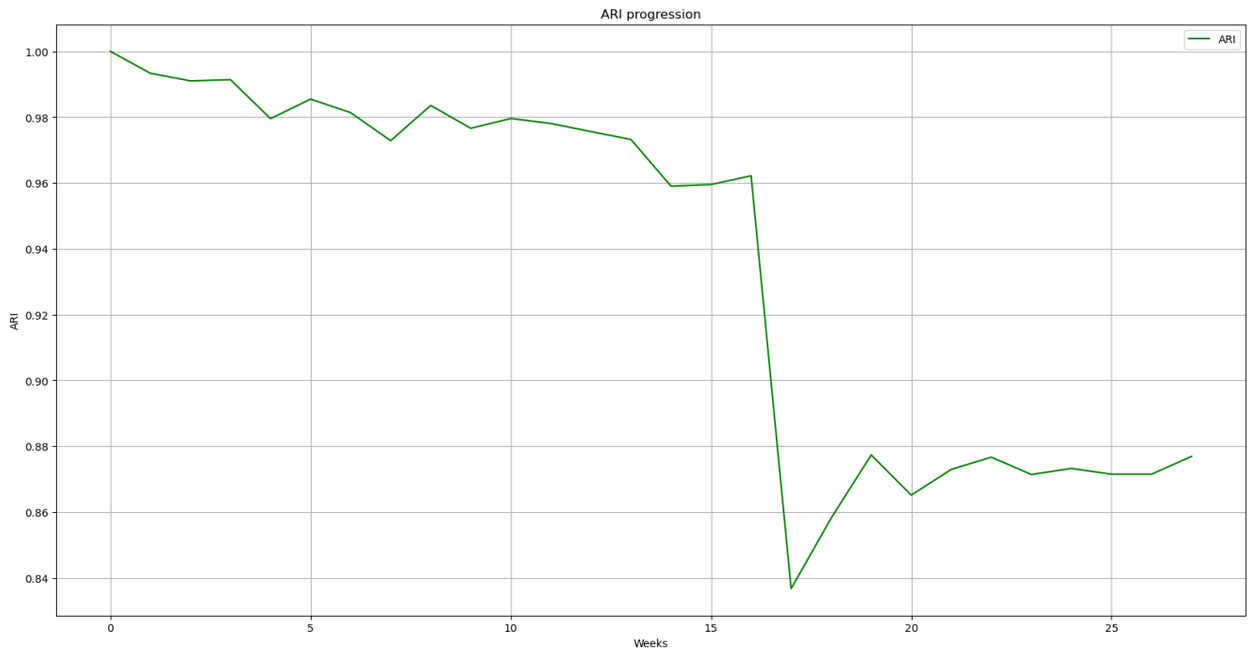
ARI Score = Adjusted Rand Index

Indique le similarité de la segmentation entre 2 partitions

En 2017, une certaine instabilité du clustering a été observée, causée par une croissance des ventes plus intense et l'afflux de nouveaux clients. Depuis 2018, la situation est devenue plus stable. La baisse soudaine à la fin du mois d'août est due au fait que nous disposons de données jusqu'en septembre 2018.

En gardant à l'esprit ce qui précède, on peut déterminer la stratégie de recyclage du modèle. Par exemple, réentraînement mensuel, puis ajustement de la fréquence en fonction des résultats et des changements de données. Ceci est particulièrement important pour que les données sur les achats au détail en ligne reflètent les variations saisonnières. La clé est de suivre la qualité du modèle au fil du temps afin de savoir quand il commence à perdre en précision.





Stabilité des segments: les démarches

```
1) entraîner le modèle sur les données de la période de base.
2) augmenter le délai de recherche d'une semaine/mois.
3) recalculer les données RFM en tenant compte des données ajoutées.
4) entraîner le modèle sur les données actuelles (kmean_current )
5) forcer les deux modèles (base(kmeans_ref) et current(current_kmeans)) à prédire les données d'entraînement
   en utilisant les données actuelles (current_rfm_values)
6) calculez l'ARI et passez à l'itération suivante.
```

```
evaluate_cluster_stability_over_time(data, initial_training_period=pd.DateOffset(months=6), freq='W', random_state=
data = data.sort_values('order_purchase_timestamp')
max_date = data['order_purchase_timestamp'].max()
start_date = max_date - initial_training_period
training_data = data[data['order_purchase_timestamp'] <= start_date]

rfm_values = calculate_rfm_values(training_data, start_date)
standardized_rfm_values = standardize_data(rfm_values)
kmeans_ref = KMeans(n_clusters=4, random_state=random_state).fit(standardized_rfm_values)

ars_scores = {}
date_range = pd.date_range(start=start_date, end=max_date, freq=freq)

for current_date in date_range:
    current_data = data[data['order_purchase_timestamp'] <= current_date]

    current_rfm_values = calculate_rfm_values(current_data, current_date)
    standardized_current_rfm_values = standardize_data(current_rfm_values)

    current_kmeans = KMeans(n_clusters=4, random_state=random_state).fit(standardized_current_rfm_values)

    ref_labels = kmeans_ref.predict(standardized_current_rfm_values)
    current_labels = current_kmeans.predict(standardized_current_rfm_values)

    ars = adjusted_rand_score(ref_labels, current_labels)

    print(f'Adjusted Rand Score for end date {current_date}: {ars}')

    ars_scores[current_date] = ars
    kmeans_ref = current_kmeans
    print(current_rfm_values.sort_values('recency'))

    current_rfm_values['Cluster'] = current_labels
    print(current_rfm_values.groupby('Cluster').describe().T)
return ars_scores
```

Améliorations à faire

En gardant à l'esprit ce qui précède, on peut déterminer la stratégie de recyclage du modèle. Par exemple, réentraînement mensuel, puis ajustement de la fréquence en fonction des résultats et des changements de données. Ceci est particulièrement important pour que les données sur les achats au détail en ligne reflètent les variations saisonnières. La clé est de suivre la qualité du modèle au fil du temps afin de savoir quand il commence à perdre en précision.


estoque online
atração de clientes
múltiplos canais de venda
coleta de pedidos
campanhas promocionais
olist
faturar pedidos
ominicanalidade
loja oficial em marketplaces
frete barato
fluxo financeiro
PDV

MERCI !