Projet 5 - 70 heures

Segmentez des clients d'un site e-commerce

Vous êtes consultant pour Olist, un site e-commerce brésilien. Les équipes marketing ont besoin de segmenter leurs clients pour optimiser les campagnes de communication.

Sommaire

1- Problématique et présentation du projet

2- Analyse Exploratoire des Données (EDA)

3- Preprocessing

4- Modeling et optimisation

5-Maintenance du modèle

6- Conclusion

Sommaire

1- Problématique et présentation du projet

2- Analyse Exploratoire des Données (EDA)

3- Preprocessing

4- Modeling et optimisation

5-Maintenance du modèle

6- Conclusion

Présentation du projet

olist, une entreprise brésilienne qui propose une solution de vente sur les marketplaces en ligne. Elle souhaite fournir à ses équipes d'e-commerce une **segmentation des clients** qu'elles pourront utiliser au quotidien pour leurs campagnes de communication.

Problématiques

Proposer une segmentation sur l'ensemble des clients qui doit être exploitable et facile à utiliser par l'équipe Marketing de Olist. Elle doit au minimum pouvoir différencier les bons et moins bons clients en termes de ***commandes*** et de ***satisfaction***.

Objectifs

- Comprendre les différents types d'utilisateurs grâce à leur comportement et à leurs données personnelles. Pour cela, des méthodes non supervisées ont été utilisées pour regrouper des clients de profils similaires.
- Fournir à l'équipe marketing une description actionable de votre segmentation et de sa logique sous-jacente pour une utilisation optimale
- Proposer un contrat de maintenance basée sur une analyse de la stabilité des segments au cours du temps.

Sommaire

1- Problématique et présentation du projet

2- Analyse Exploratoire des Données (EDA)

3- Preprocessing

4- Modeling et optimisation

5-Maintenance du modèle

6- Conclusion

Découverte des données

- Olist vous fournit une base de données anonymisée comportant des informations sur l'historique de commandes, les produits achetés, les commentaires de satisfaction, et la localisation des clients depuis janvier 2017.
- Les données sont à télécharger sur Kaggle : https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce
- Les données fournies par Olist sont réparties dans plusieurs tables :

Tables clients

Customers (99441 , 5)
Geolocation (1000163 , 5)

Tables commandes

Orders (99441 , 8)
Order_items (112650 , 7)
Order payment (103886,5)
Order review (99224 , 7)

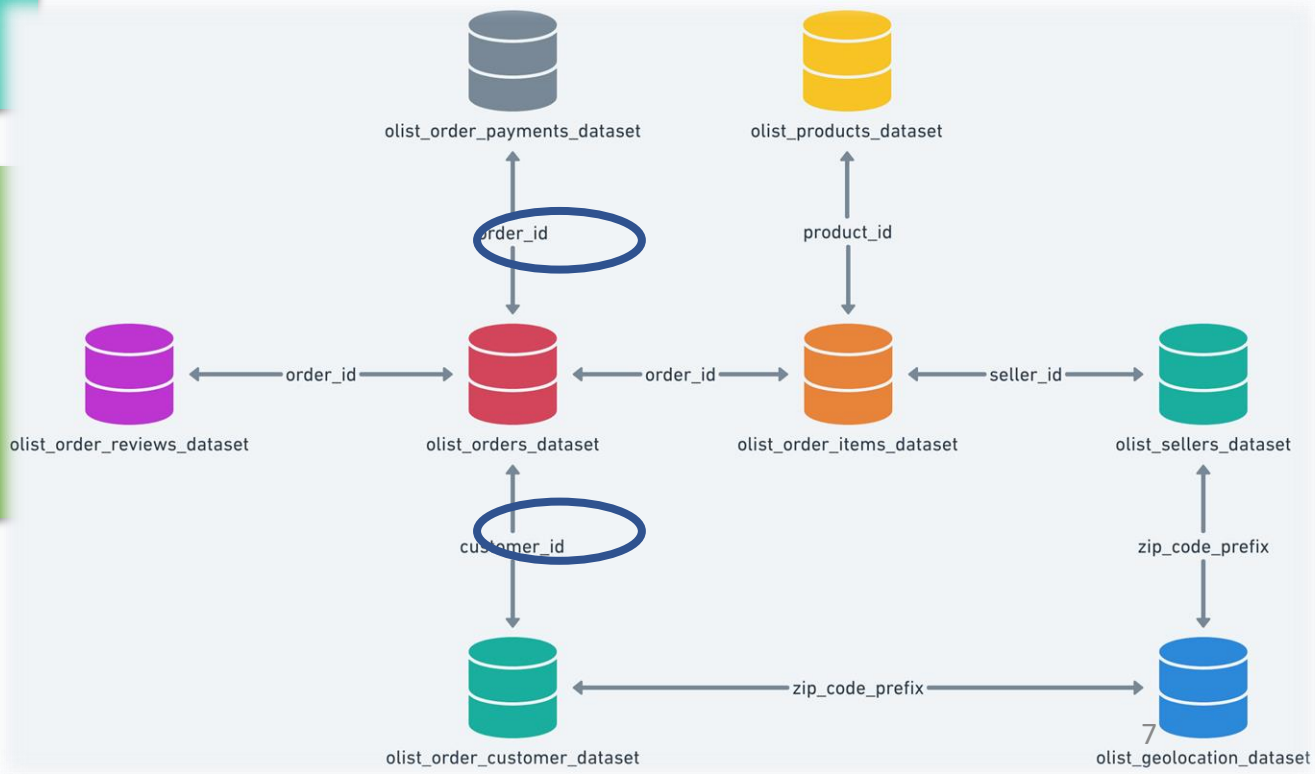
Tables produits

Products (32951 ,9)
Product_category_translation (71 , 2)

Tables vendeurs

Sellers (3095 , 4)

Les différents jeux de données sont reliées par des « ***Primary key*** » afin de faire la jointure des fichiers.



Les différentes tables ont été rassemblées dans un seul dataframe ayant un client par lignes en utilisant les commandes ***groupby*** et ***merge*** de pandas.

Orders

Order_items

Order_payment

Groupby('order_id) pour avoir Nombre d'échéance par comm

Order_payment

Order_review

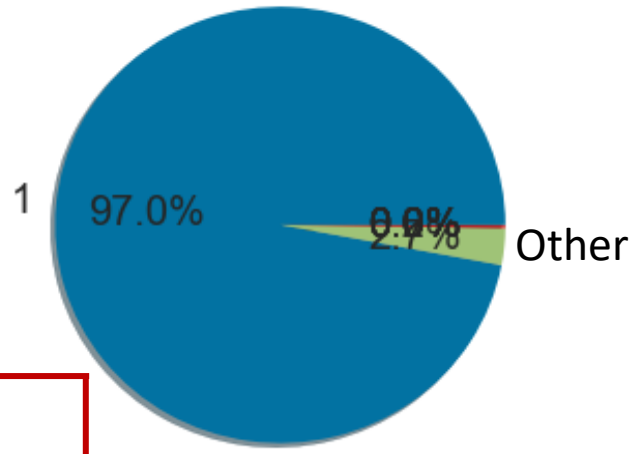
Groupby('order_id) pour avoir score moyen par commande

Order_review

Customers

Products

Category_translation



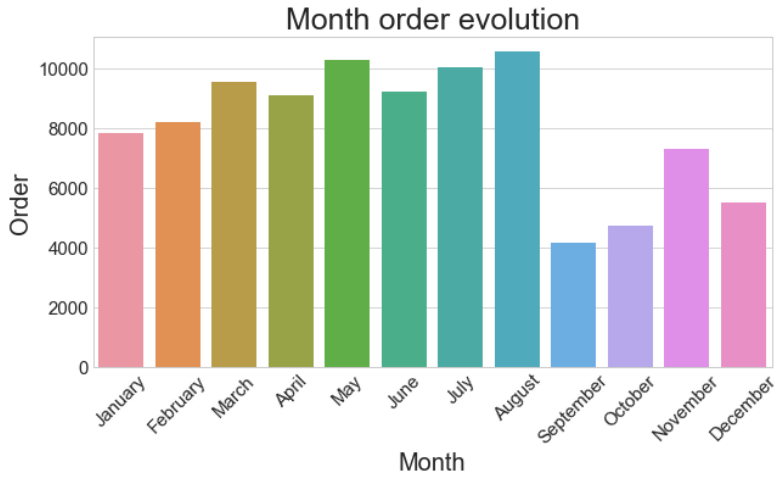
Dataset
107820 lignes
20 colonnes

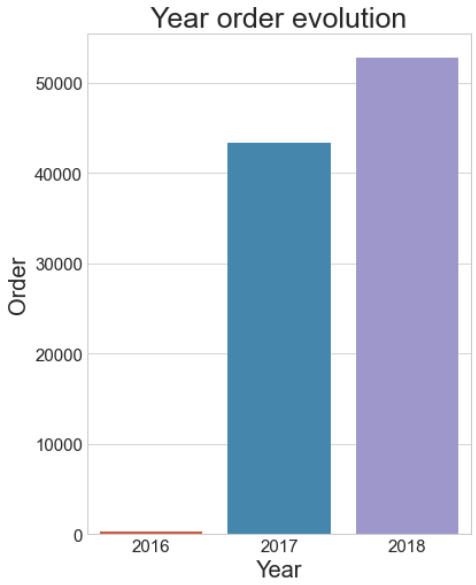
La majorité des client ont passé une seule commande.

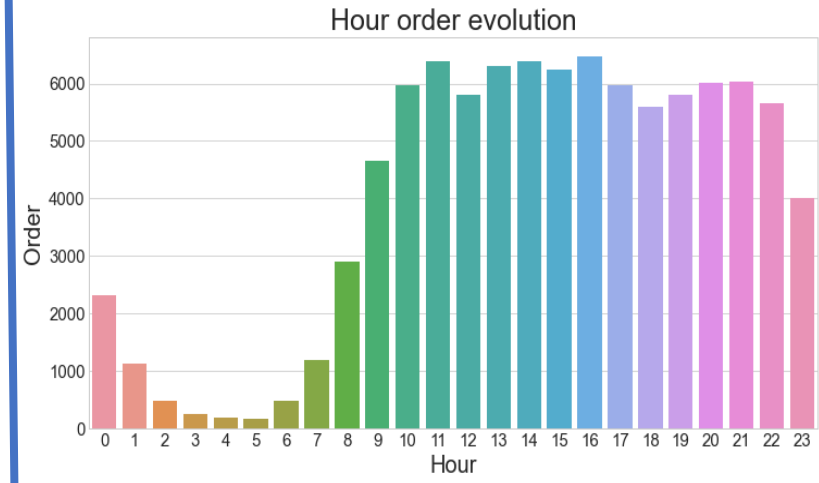
Nous avons effectué une analyse exploratoire des données pour chaque dataset.

L'analyse temporelle du dataset orders montre que :

- Les clients semblent commander plus en semaine qu'en week-end.
- Les commandes évoluent tout au long de l'année avec une baisse pour les 4 dernier mois.
- Les commandes s'étalent du 03/10/2016 jusqu'à 29/08/2018, la raison pour laquelle le nombre de commandes en 2016 et très inférieurs par rapport à 2017 et 2018.





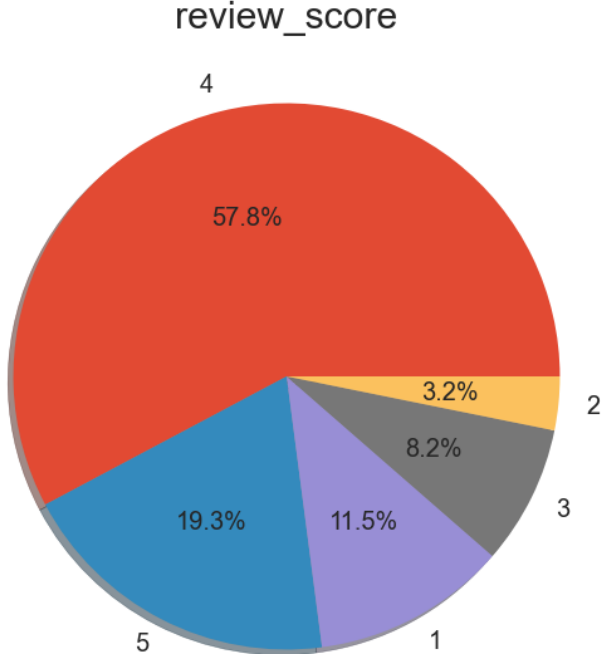


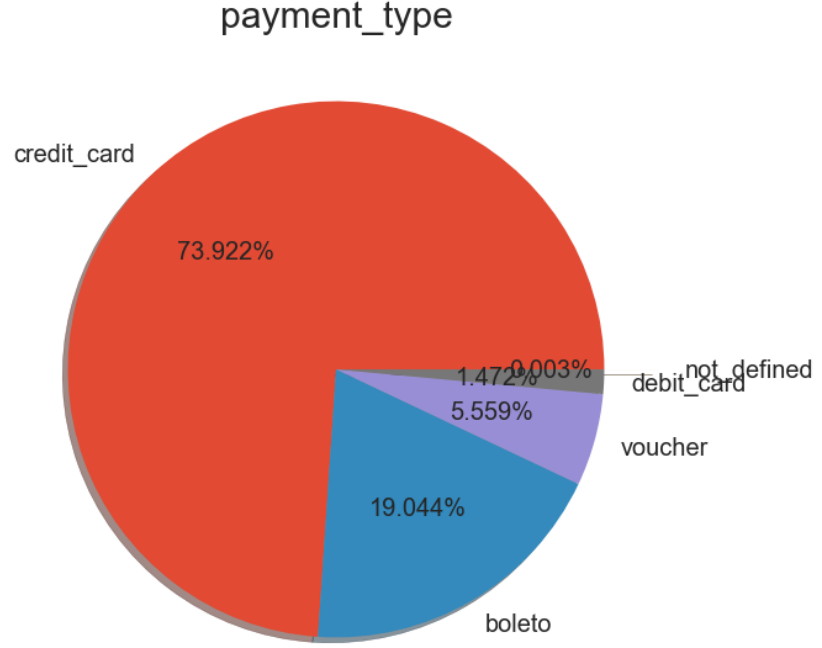
L'analyse des datasets order_reviews, order_payment et customers montre que :

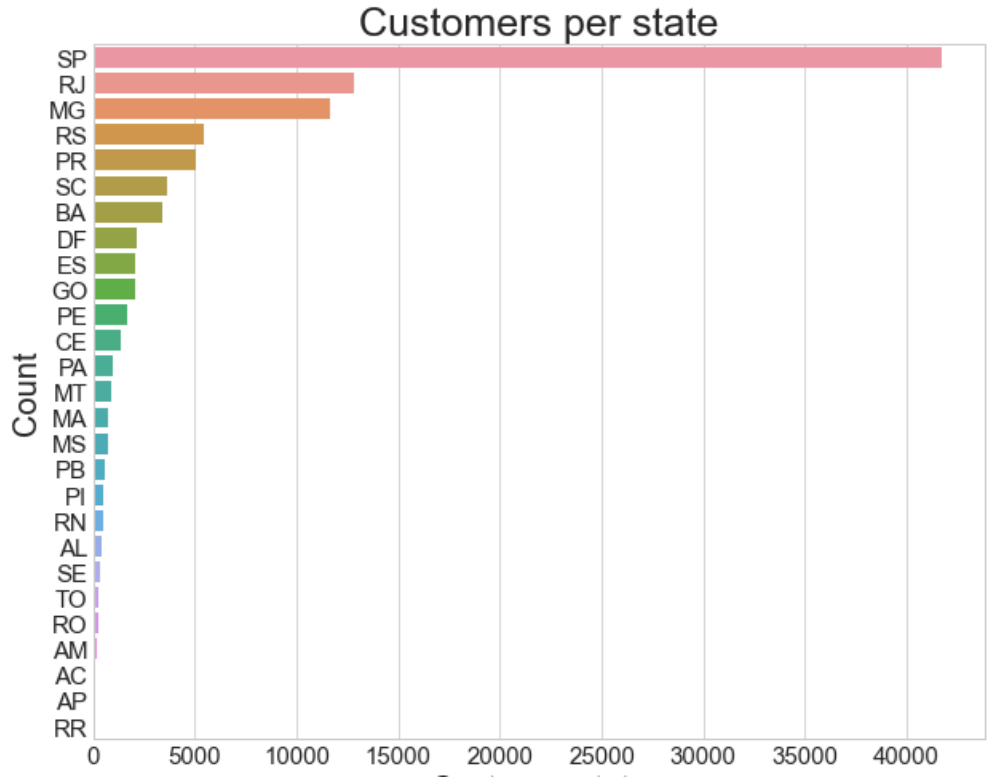
- Sao Paulo est la ville brésilienne où le nombre de commandes est le plus élevé suivie par Rio de Janeiro puis Minas Gerais.

- Plus de 73% Les clients utilisent la carte de crédit comme moyen de paiement.

- La majorité des clients sont satisfaits par le site de vente en ligne Olist



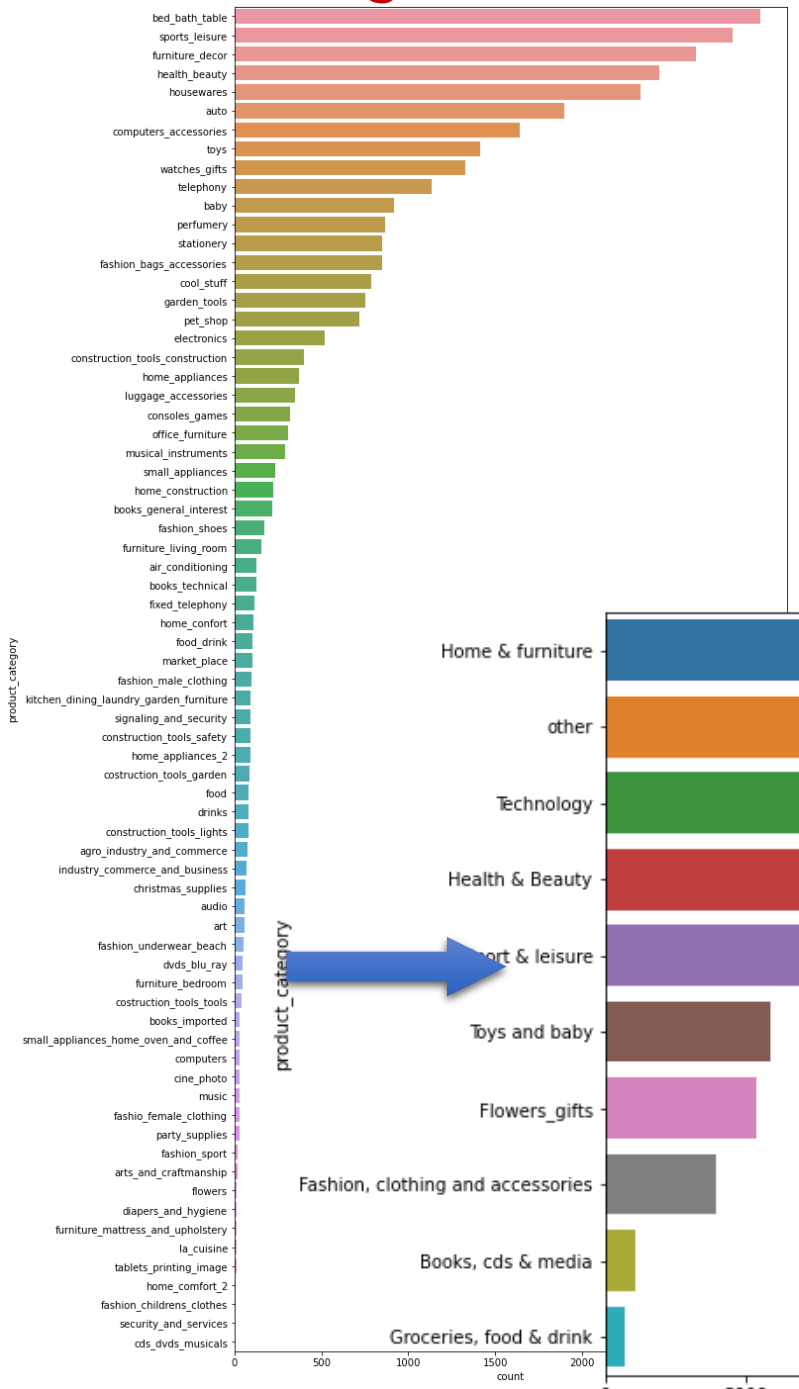


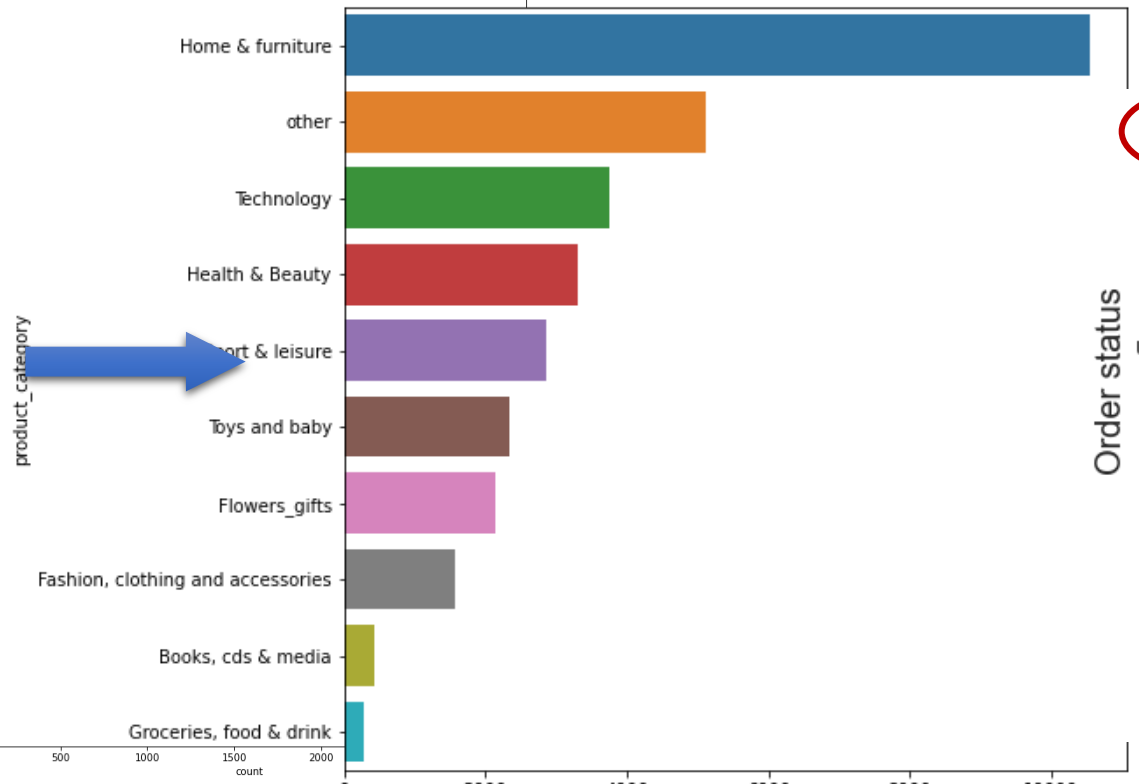


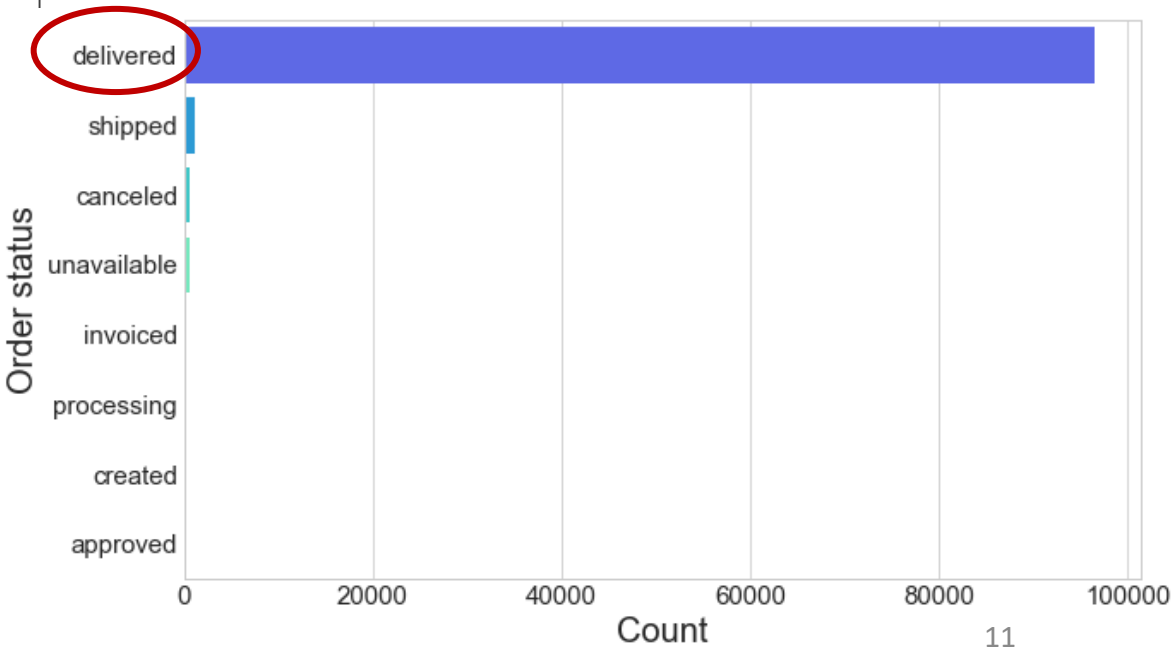
Data cleaning

Plusieurs étapes de nettoyage ont été faites sur les jeux de données :

- Plusieurs colonnes redondantes ont été supprimées.
- Les datasets *Products et product_category_translation* contiennent plus de 70 catégories. Ces dernières ont été regroupées en 10 grandes familles. Cette classification des produits vendus en ligne est données sur ce lien https://www.statista.com/statistics/311406/us-online-shopping-categories-gender/.
- La catégorie Home et Furniture est largement présentées.
- Nous avons garder uniquement les clients qui ont été délivrés dans le dataset *df_orders*.







Sommaire

1- Problématique et présentation du projet

2- Analyse Exploratoire des Données (EDA)

3- Preprocessing

4- Modeling et optimisation

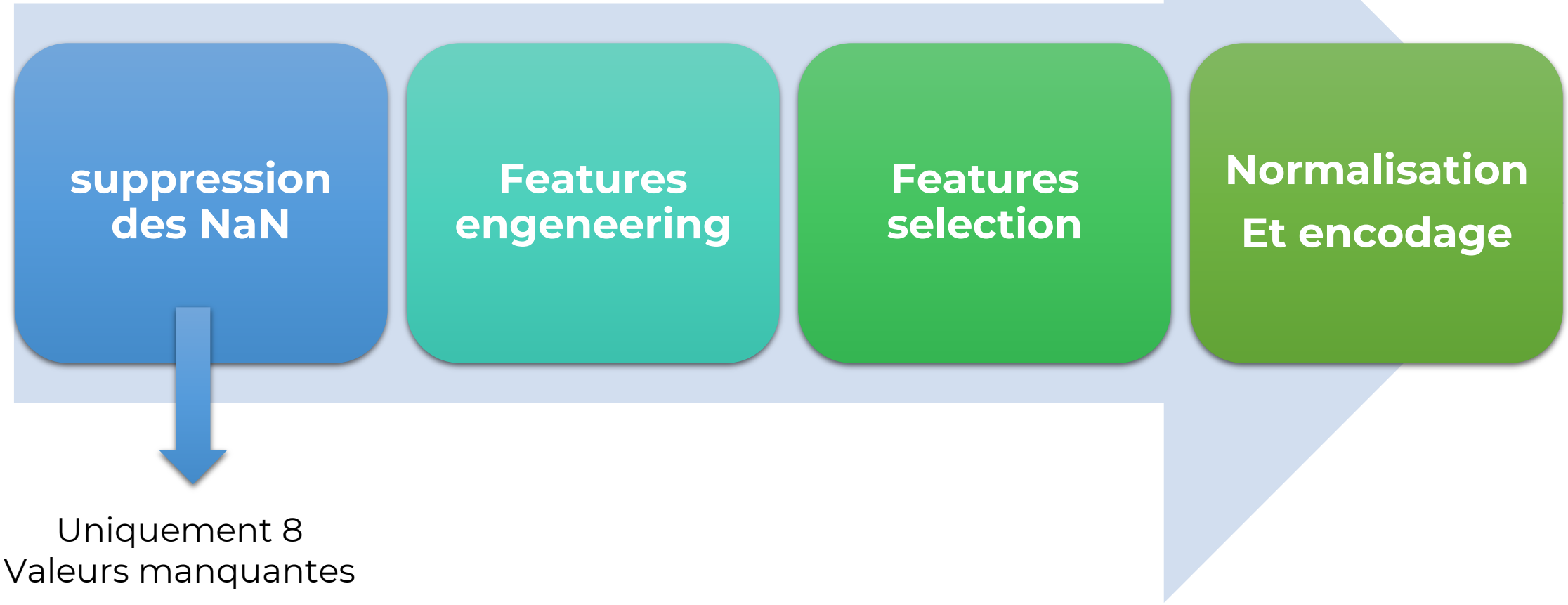
5-Maintenance du modèle

6- Conclusion

Preprocessing

Dataset initial

Pipeline de preprocessing



Uniquement 8
Valeurs manquantes

Dataset prêt pour la modélisation

Features engineering

Le dataset final après feature engeneering est basé sur l'identifiant unique du client. Il regroupe :

Variables totales	Variables moyennes	Variables catégorielles	Variables mensuelles	Variables géographiques
-Frequency -total_item -Monetary -total_freight -freight_ratio	-mean_review_score -order_mean_delivery_delay(day) -Recency -mean_delay_between_orders -Order_mean_installment...	-ratio_price/category -ratio_item/category	-ratio_price/month -ratio_item/month -ratio_order/month	-ratio_price/state

À la fin de cette étape, le dataset contient 91481 lignes et 101 colonnes.

Features selection

- supprimer les features fortement corrélées (Pearson > 0.85).
- Après cette sélection, nous avons obtenu 61 colonnes.

	total_orders	total_item	total_products_payment	total_freight_payment	total_turnover	freight_ratio	order_mean_item
total_orders	1.0	0.081	0.11	0.21	0.12	-0.012	-0.15
total_item	0.081	1.0	0.16	0.44	0.2	0.071	0.96
total_products_payment	0.11	0.16	1.0	0.42	1.0	-0.42	0.13
total_freight_payment	0.21	0.44	0.42	1.0	0.5	0.11	0.38
total_turnover	0.12	0.2	1.0	0.5	1.0	-0.39	0.16
freight_ratio	-0.012	0.071	-0.42	0.11	-0.39	1.0	0.074
order_mean_item	-0.15	0.96	0.13	0.38	0.16	0.074	1.0
order_products_mean_price	-0.011	0.15	0.99	0.39	0.98	-0.42	0.15
order_freight_mean_price	0.0012	0.42	0.41	0.97	0.48	0.11	0.42

Encodage & normalisation

- Cette étapes est essentielle pour avoir des ordres de grandeurs proches des différentes variables.
- Toutes les variables ont été normalisées avec la méthode *StandardScaler()* utilisant la moyenne (**μ**) et l'écart type (**σ**).
- Les variables ratio n'ont pas été normalisées puisqu'elles sont comprises entre 0 et 1.

$$z = \frac{x - \mu}{\sigma}$$

Sommaire

1- Problématique et présentation du projet

2- Analyse Exploratoire des Données (EDA)

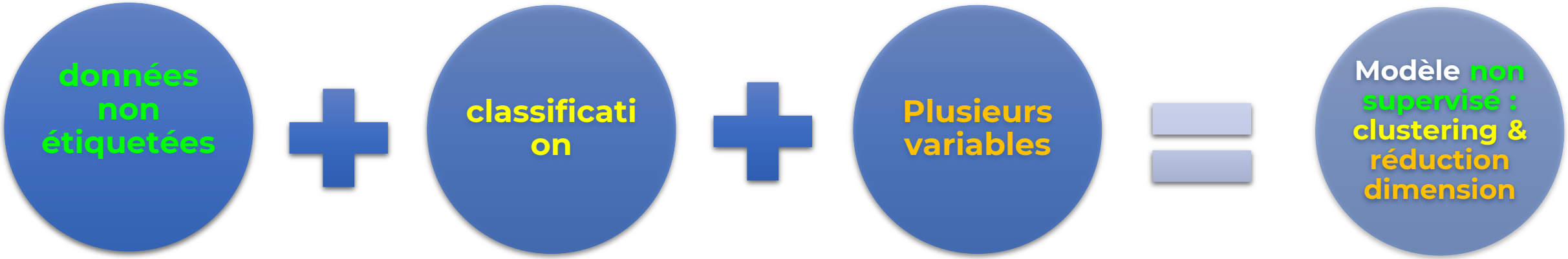
3- Preprocessing

4- Modeling et optimisation

5-Maintenance du modèle

6- Conclusion

Présentation des modèles



Modèles testés

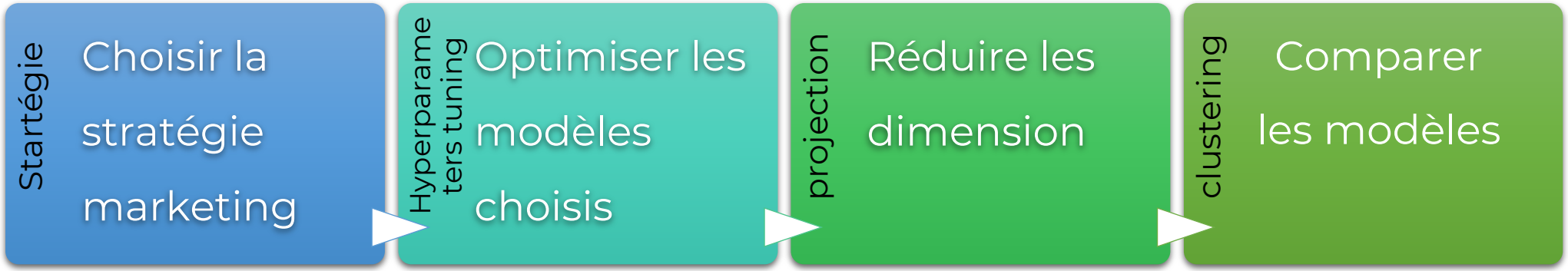
Clustering

- KMeans
- DBSCAN
- AgglomerativeClustering

Réduction dimensionnelle

- PCA
- t-SNE

Démarche de modélisation



Stratégie de segmentation

Quand on veut vendre « tout à tout le monde », on ne vend rien à personne.

Pour vendre de façon efficace, nous devons « segmenter » notre cible. Segmenter les clients c'est diviser la base clients en groupe d'individus similaires et pertinents pour une stratégie marketing. Cela permettra ainsi :

- d'affiner votre compréhension des clients et d'identifier leurs attentes.
- de mieux cibler votre communication et vos offres marketing et commerciales.
- d'optimiser l'allocation des ressources de votre entreprise grâce à une meilleure priorisation.

Il existe diffèrent types de segmentation :

- **La segmentation démographique** (âge, genre, le revenu, le niveau d'études, le statut marital…)
- **La segmentation géographique** (lieu de résidence ou de consommation ou de travail…)
- **La segmentation psychographique** (mode de vie, les hobbies, les intérêts, ou encore les traits de personnalité…)
- **La segmentation comportementale** (la fréquence d'achat, les marques préférées, la sensibilité au prix, la loyauté…)
- **La segmentation RFM - Récence, Fréquence, Montant**
  - La **récence** permet de situer le dernier achat dans le temps.
  - La **fréquence** indique le nombre de fois où votre client a fait une transaction pendant une période donnée.
  - Le **montant** correspond à la somme ou moyenne des dépenses de votre client.

Dans ce projet, nous nous sommes focaliser sur ***la segmentation RFM et comportementale.***

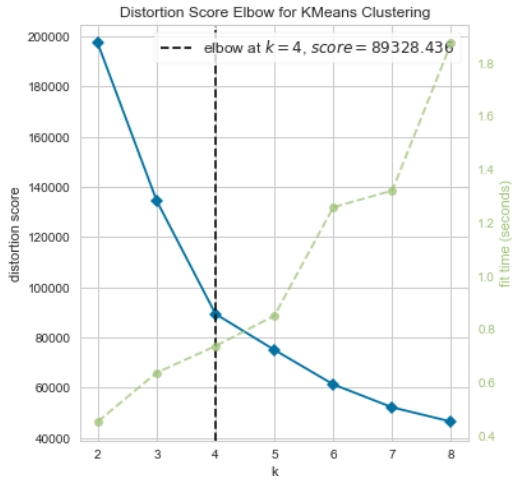
Stratégie RFM : Clustering with Kmeans

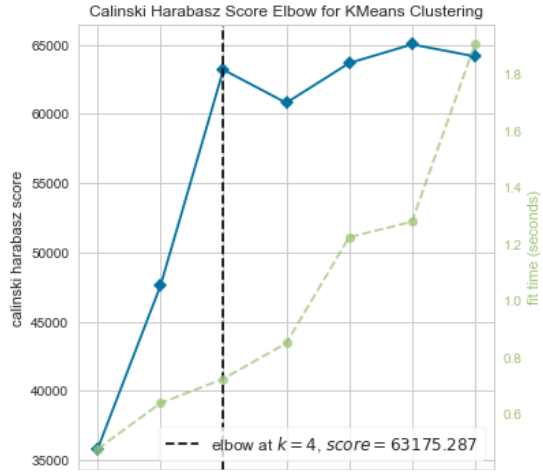
- La **récence** : nombre des jours écoulés depuis dernier achat.
- La **fréquence** : nombre total des commandes
- Le **montant** : somme des dépenses totales du client sans transport..

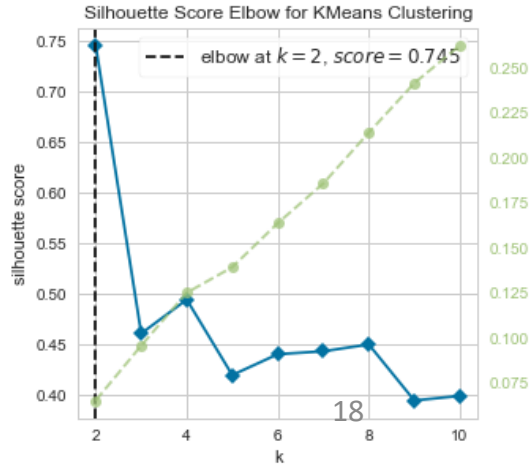
Optimisation du nombre de clusters k

- Afin de déterminer le nombre optimal des clusters, la méthode de coude a été utilisée avec différentes métriques (distorsion, Calinski Harabasz et silhouette). En réalisant une itération de l'algorithme Kmeans sur un intervalle de k, les métriques sont affichées sur les courbes.
- Nous avons aussi calculer le nombre des clients par groupe pour différentes valeurs de k.
- Le point d'inflexion de la courbe détermine la valeur de k hypothétique.
- Après analyse des courbes et du tableau, nous avons opter pour la valeur ***k=3***.

```
the number of customers per cluster in 2 clusters is:
 0    88761
1     2712
Name: 2 clusters, dtype: int64
____________________________________________________
the number of customers per cluster in 3 clusters is:
 0    50942
1    37819
2     2712
Name: 3 clusters, dtype: int64
____________________________________________________
the number of customers per cluster in 4 clusters is:
 2    49634
0    36736
1     2684
3     2419
Name: 4 clusters, dtype: int64
____________________________________________________
the number of customers per cluster in 5 clusters is:
 4    33559
1    32711
0    20403
2     2683
3     2117
Name: 5 clusters, dtype: int64
____________________________________________________
the number of customers per cluster in 6 clusters is:
 5    32680
0    31644
4    19903
1     3994
2     2684
3      568
Name: 6 clusters, dtype: int64
____________________________________________________
```

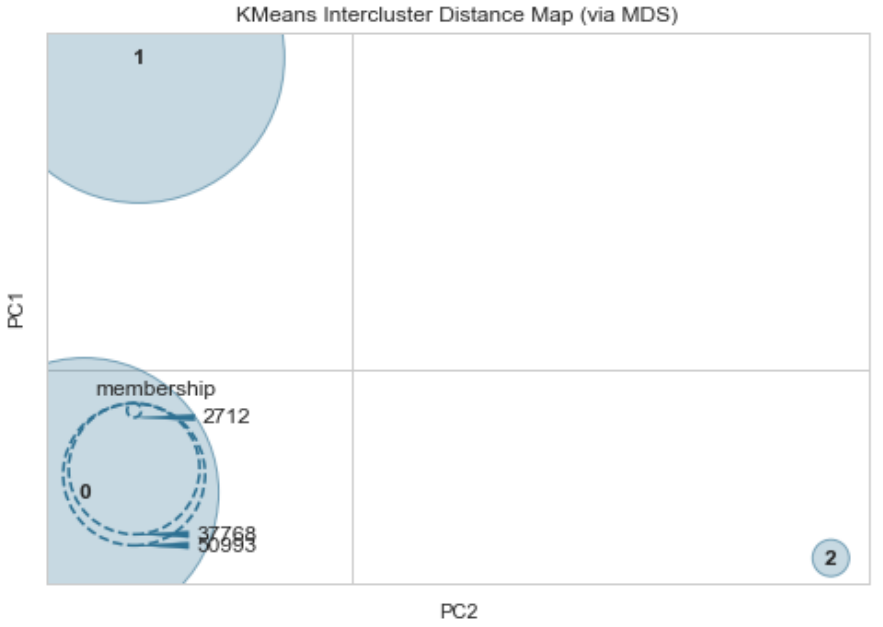


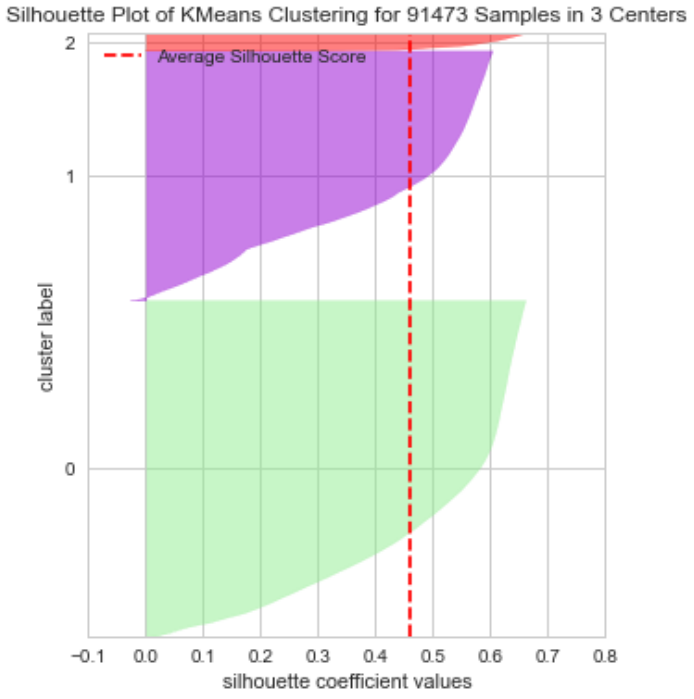


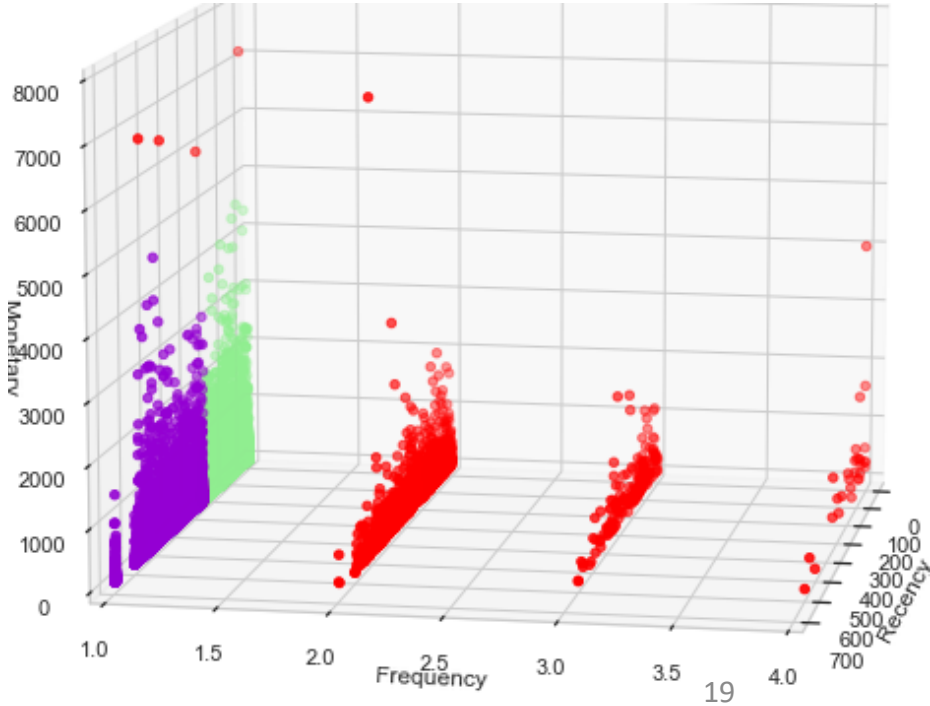


Stratégie RFM : Clustering with Kmeans

- Le graphe« Silhouette plot » permet de visualiser la densité et la séparation des différents clusters avec un score de silhouette moyen de **0.46** .
- Le graphe « Intercluster Distance » montre la projection des cluster sur les 2 premières composatntes principales de la MDS(Multidimensional Scaling).
- Les groupes sont globalement bien réparties et séparés.
- Le cluster 2 (en rouge) est moins dense que les autres groupes. Probablement, ça représente les 3% des clients qui ont plus d'une commande. C'est très clair sur la présentation en 3D.
- Nous devons à présent analyser les différents clusters pour déterminer leurs caractéristiques.







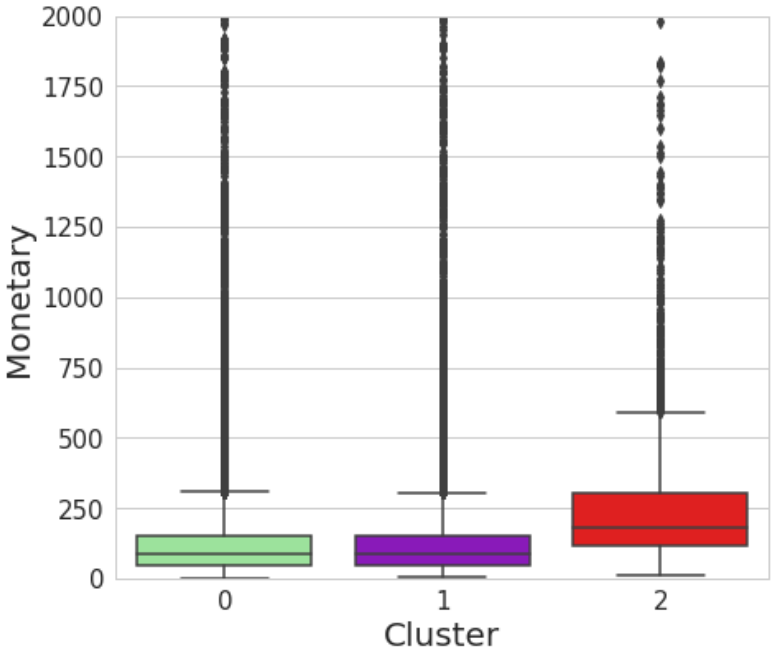
Stratégie RFM : Clustering with Kmeans

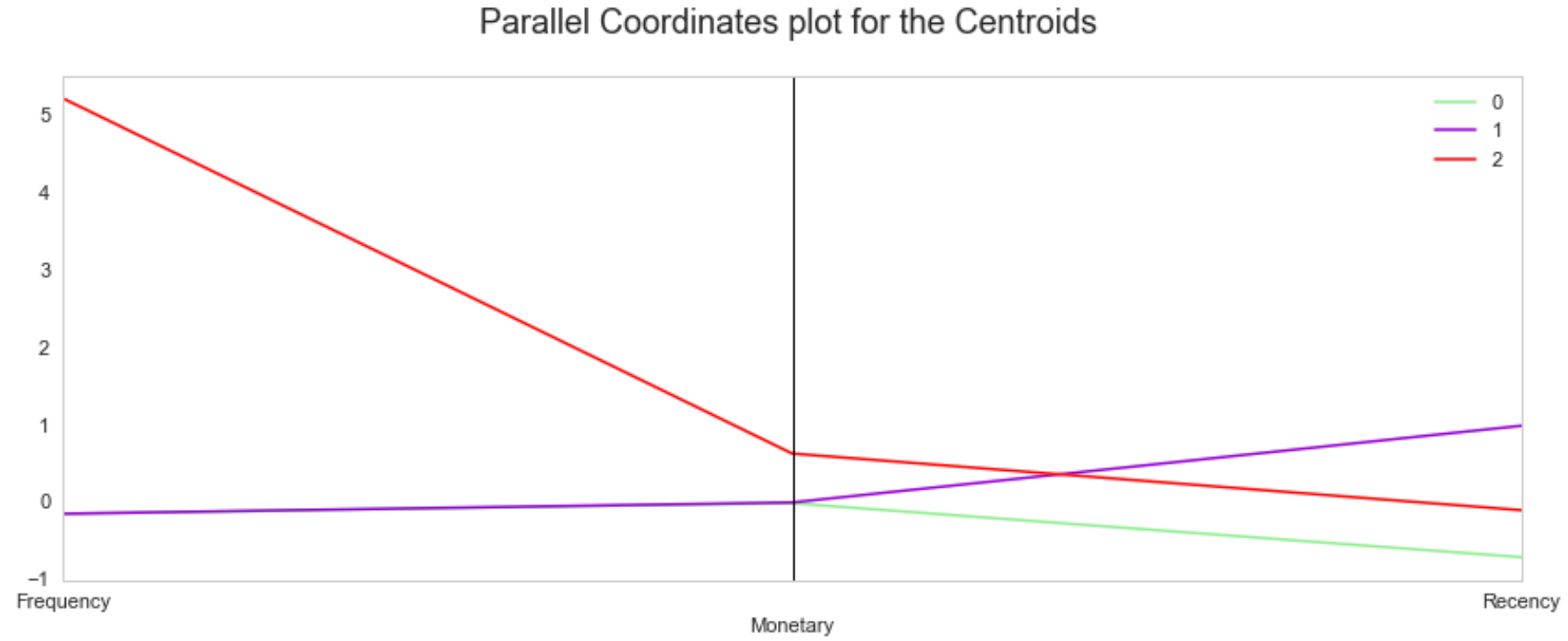
Analyse des clusters

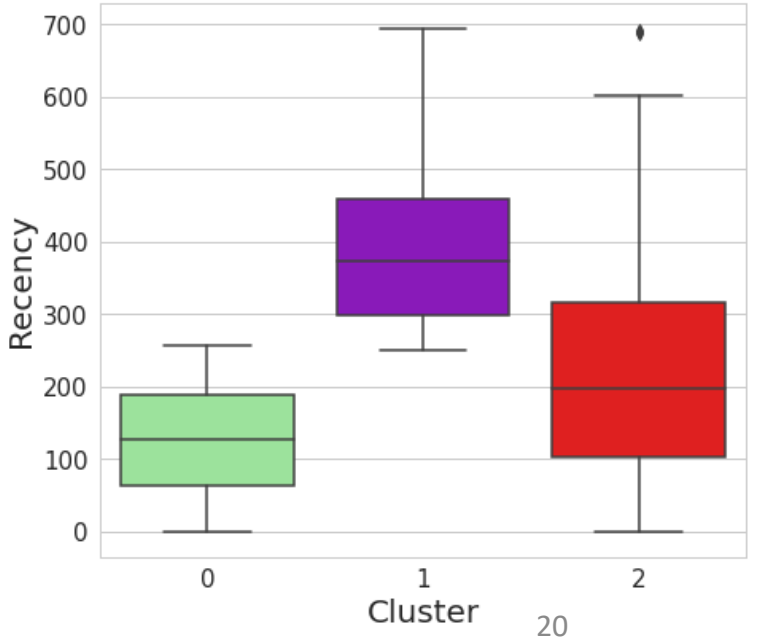
- Le cluster 2 représente les clients qui ont passé plus d'une commande. Ces clients représentent uniquement 3%.

- Les clusters 0 et 1 sont les clients qui ont une seule commande. Mais ceux du groupe 0 sont plus récents. Les clients du cluster 1 sont anciens.

cluster	Recency	Frequency	Monetary
0	126.679616	1.000000	136.396591
1	386.125836	1.000000	138.599921
2	219.393068	2.108776	274.222334





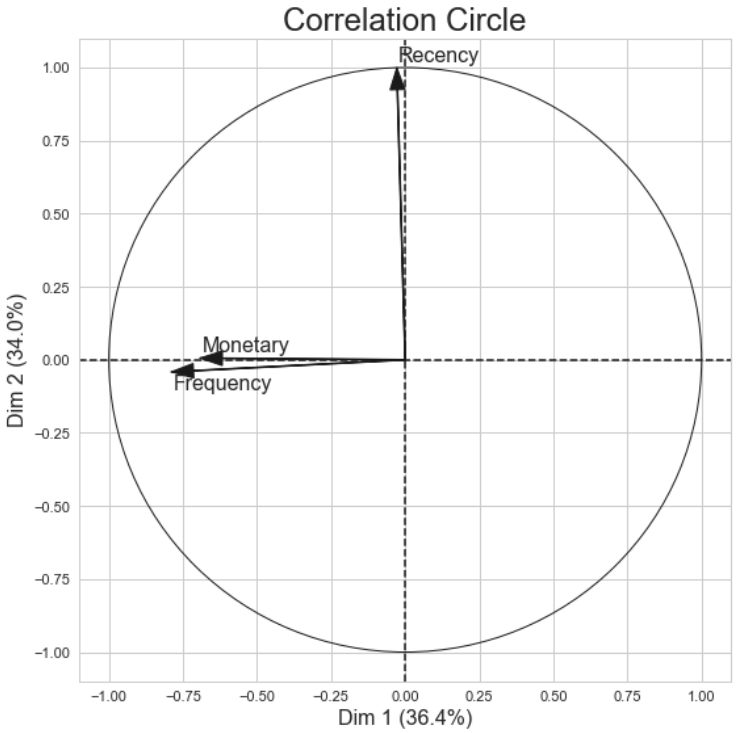


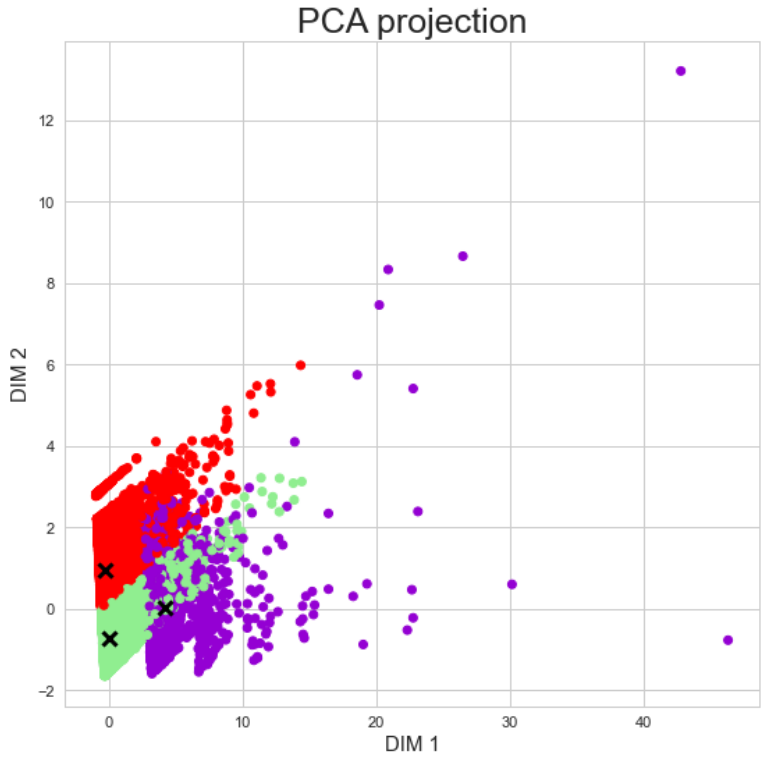
Stratégie RFM : Clustering with Kmeans

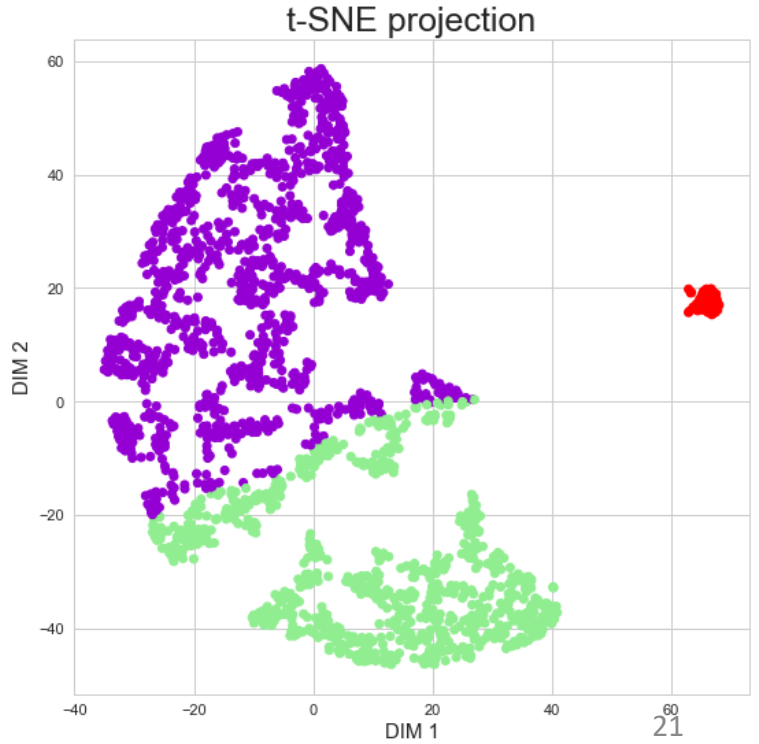
Réduction dimensionnelle

Recency est fortement corrélée avec DIM2 alors que DIM1 représente les variables Monetary et Frequency.

La projection des clusters avec t-SNE est plus intéressante que celle avec PCA. En effet les clusters sont bien séparés et denses avec la méthode t-SNE.







Stratégie RFM : Clustering with Kmeans

Stabilité de l'algorithme KMeans

KMeans stability

Iteration	FitTime	Inertia	Homo	ARI	AMI
Iter 0	0.033s	22960	0.999	1.000	0.999
Iter 1	0.045s	24439	1.000	1.000	1.000
Iter 2	0.038s	27405	0.993	0.997	0.992
Iter 3	0.048s	26764	0.986	0.993	0.986
Iter 4	0.032s	28284	0.997	0.999	0.997
Iter 5	0.031s	26822	0.996	0.998	0.996
Iter 6	0.034s	28571	0.995	0.998	0.995
Iter 7	0.047s	23959	0.989	0.995	0.989
Iter 8	0.053s	28402	0.988	0.995	0.989
Iter 9	0.052s	25587	1.000	1.000	1.000

- L'algorithme Kmeans est initialisé avec la méthode K-Means++, ce qui réduit les effets aléatoires de l'initialisation des centroïdes.

- La stabiloté à l'initialisation a été testée en entrainant plusieurs fois le modèle sans fixer le random_state.

- Plusieurs métriques ont été relevées :
  - **FitTime :** durée d'entrainement du modèle
  - **Inertia :**
  - **Homo :** Homogeinity
  - **ARI :** Adjested Rand Index
  - **AMI :** Adjusted Mutual Info

- Les différentes itérations montrent des inerties proches, une bonne homogéeité et des scores ARI et AMI proches de 1.

- Nous pouvons en déduire que le modèle Kmeans est stable à l'initialisation.

Stratégie RFM : Clustering with DBSCAN

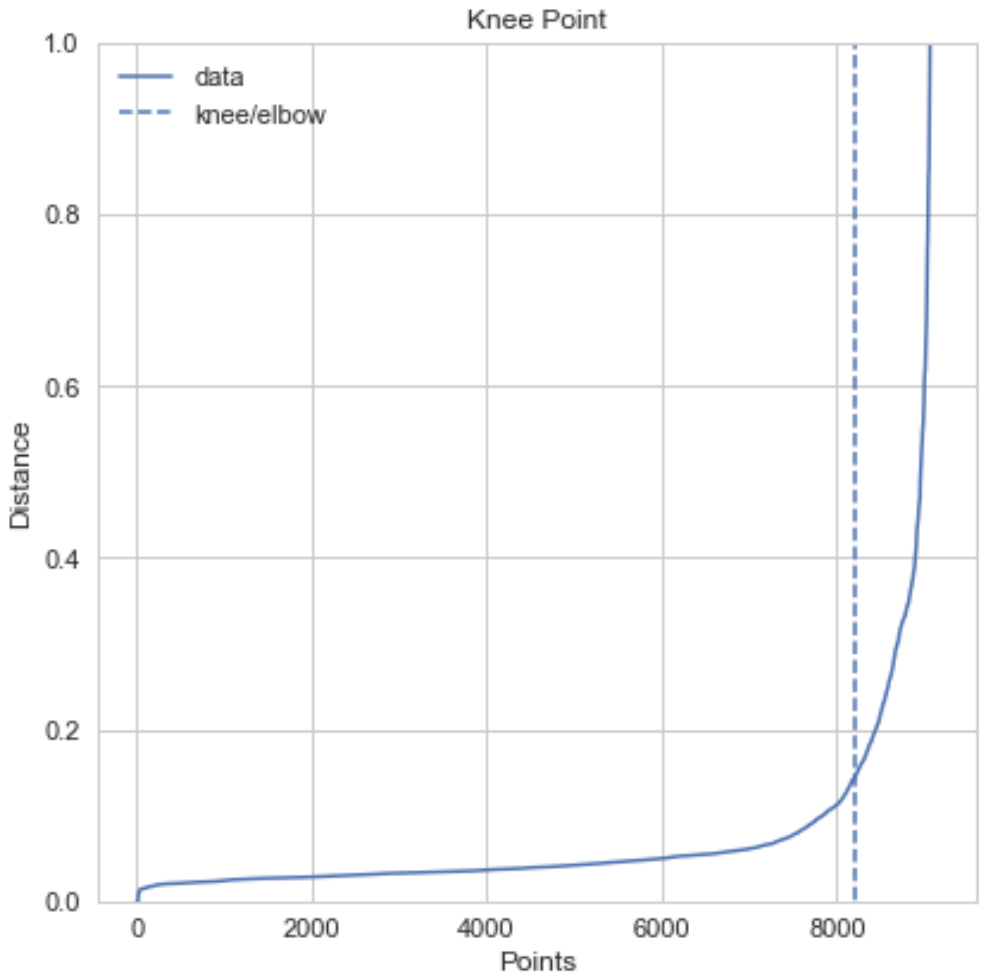
Optimisation de la distance eps

- Le DBSCAN est un algorithme simple qui définit des clusters en utilisant l'estimation de la densité locale.

- Epsilon quantifie une mesure du voisinage. Deux points sont voisins quand ils sont à une distance plus petite que epsilon l'un de l'autre.

- L'idée est de calculer la moyenne des distances de chaque point à ses k plus proches voisins. La valeur de k sera précisée et correspond à **min_samples = 40**. Ensuite, ces k-distances sont tracées dans un ordre croissant. Le but est de déterminer le « knee », qui correspond au paramètre eps optimal. Un coude correspond à un seuil où un changement brusque se produit le long de la courbe de k-distance.

- La valeur optimale est **eps = 0.146**



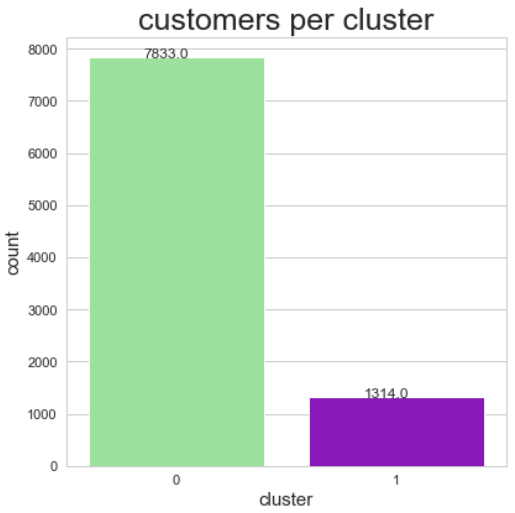
Stratégie RFM : Clustering with DBSCAN

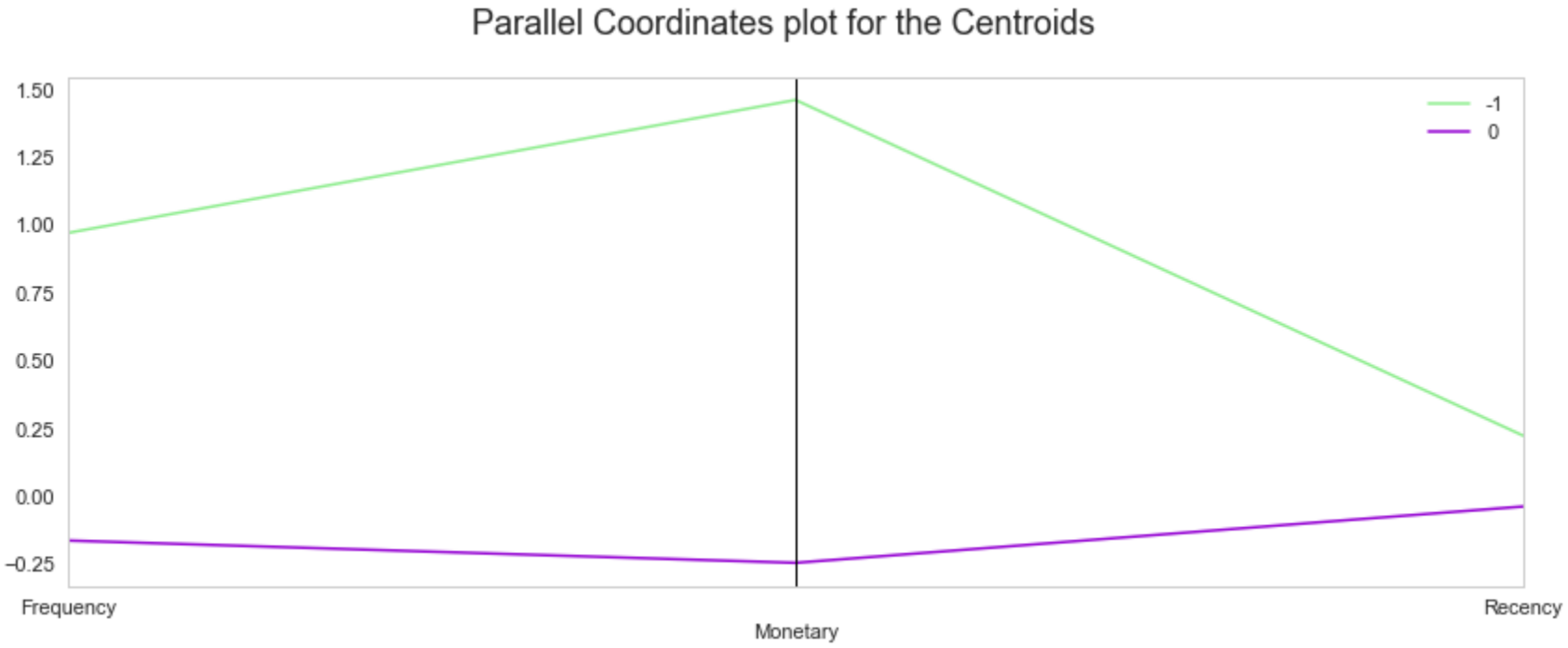
Analyse des clusters

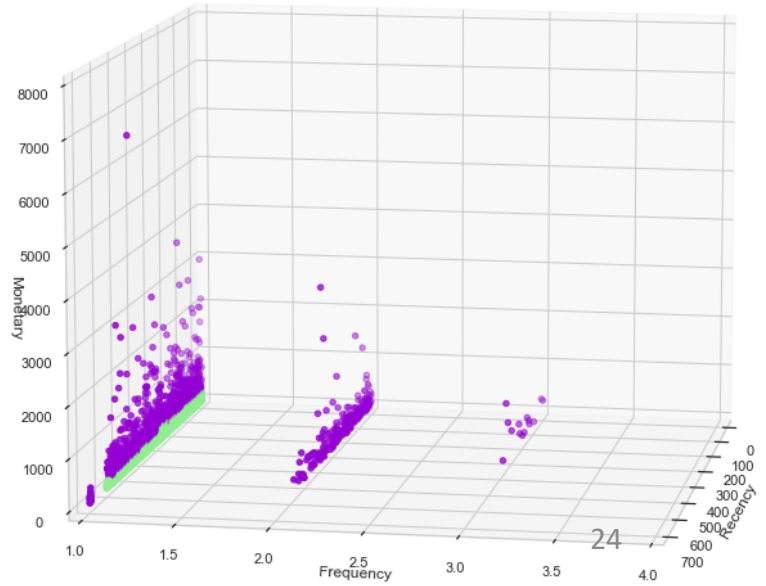
- L'algorithme DBSCAN a aboutit a 2 clusters uniquement. Ce groupement est basé surtout sur les montants dépensés par les clients.

- Le clustering n'a pas pris en considération le nombre de commandes et le temps écoulés depuis le dernier achat

cluster	Frequency	Monetary	Recency
0	1.000000	89.705944	230.519214
1	1.229833	455.302595	270.662861







Stratégie RFM : Hierarchical clustering

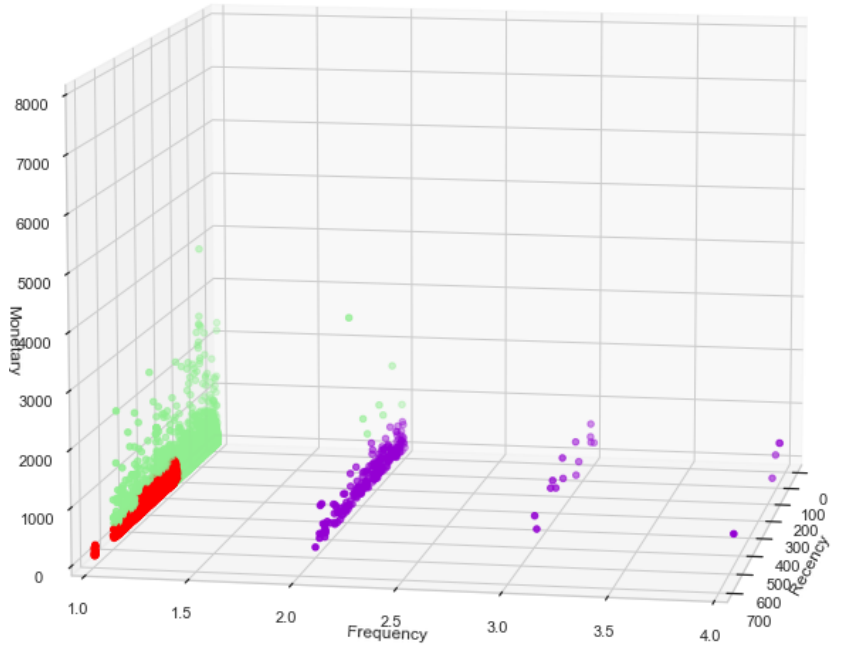
- Le regroupement hiérarchique commence par traiter chaque observation comme un groupe distinct. Ensuite, il exécute à plusieurs reprises les deux étapes suivantes :

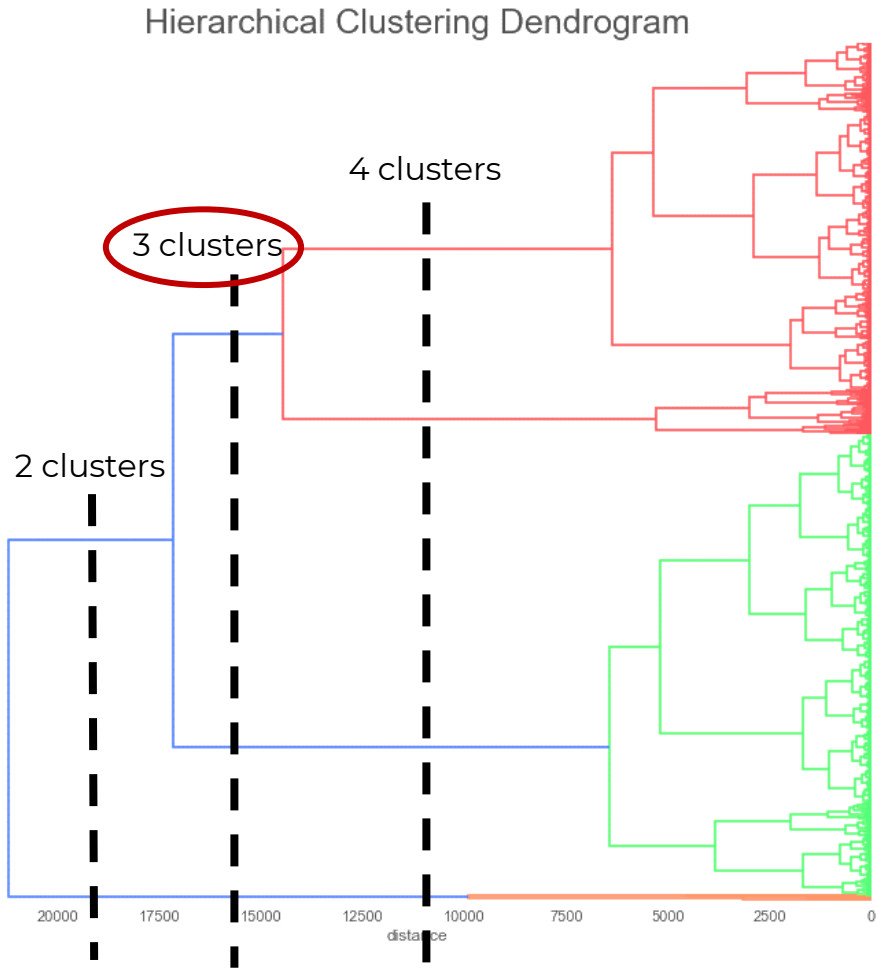
Identifier les deux clusters les plus proches.

Fusionner les deux clusters les plus similaires.

Ce processus itératif se poursuit jusqu'à ce que tous les clusters soient fusionnés.

- Le Dendrogramme est donc **le type de diagramme en arborescence** que l'on utilise pour présenter le clustering hiérarchique, à savoir les relations entre des ensembles de données similaires.
- À partir du dendrogramme, nous avons opté pour 3 clusters.





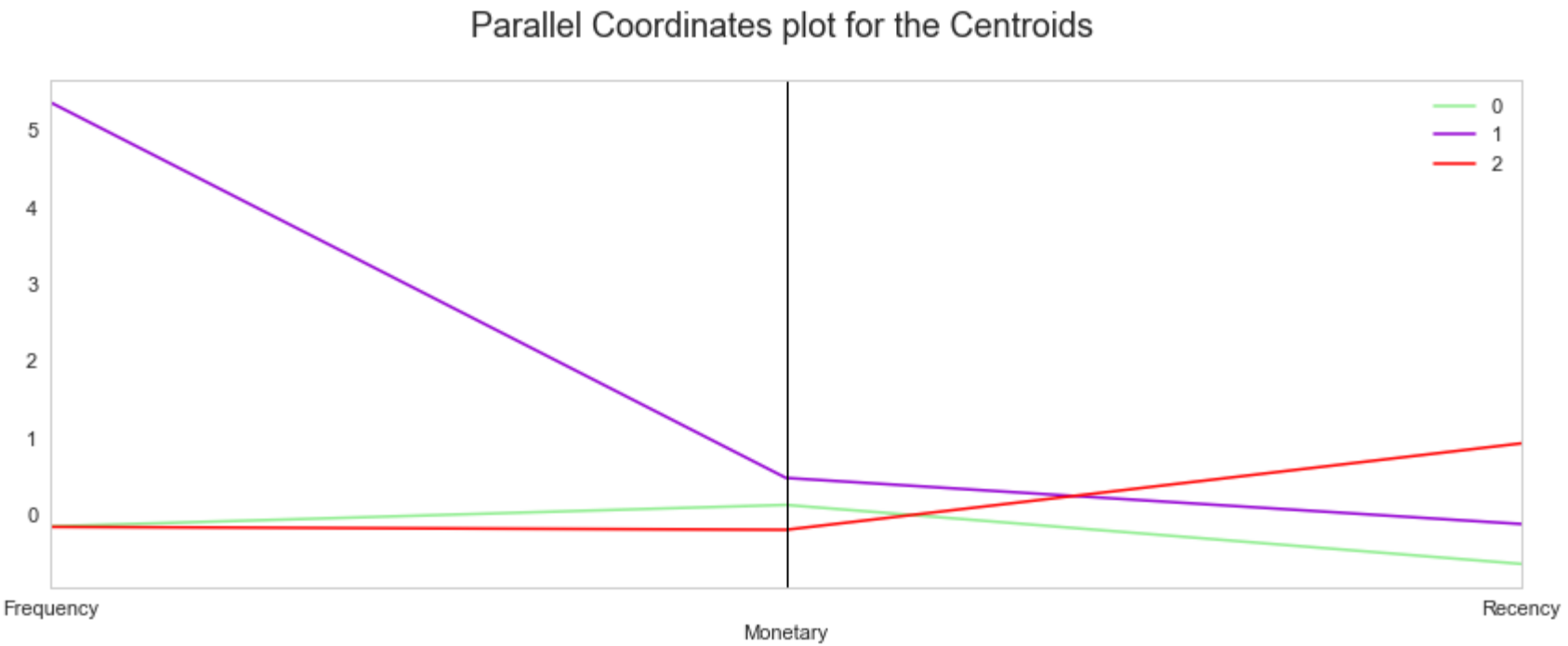
Stratégie RFM : Hierarchical clustering

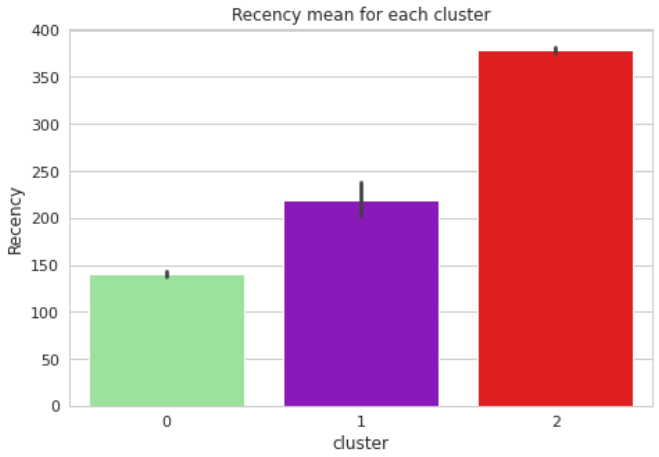
Analyse des clusters

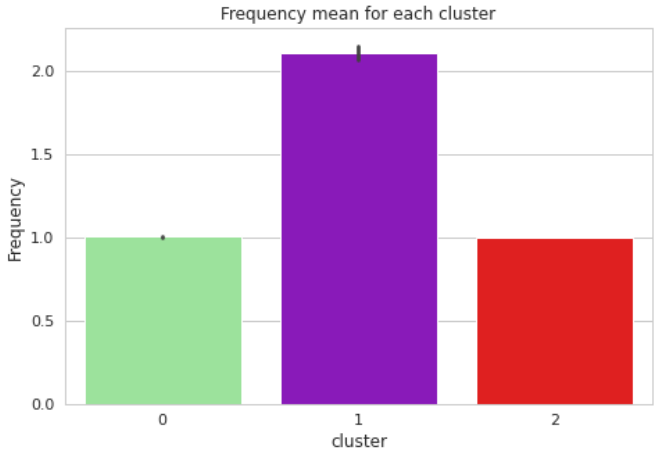
- Les cluster 0 et 2 ont la même fréuences avec une différence au niveau de l'ancienneté correspond a des clients très anciens ayant une seule commande.

- Le cluster 1 correspond aux clients ayant plus de deux commandes.

- Le clustering hiérarchique est similaire à celui donné par KMeans





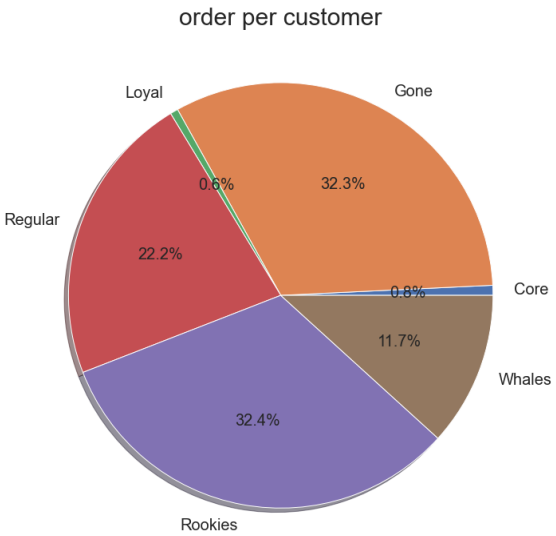


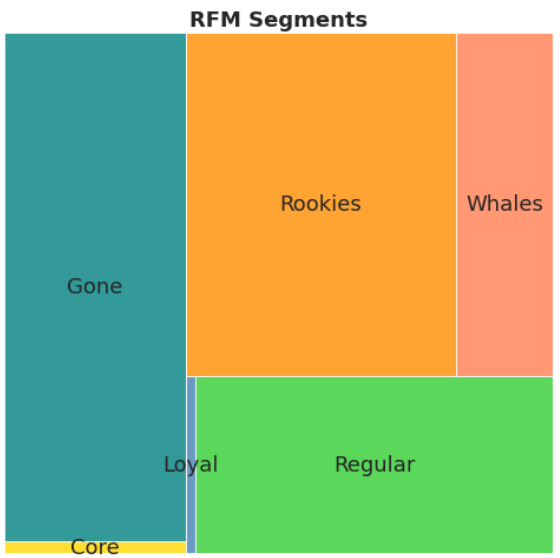
Stratégie RFM : Méthode de scoring

Le lead scoring consiste à calculer et assigner à chaque prospect un nombre de points (score) pour refléter leur potentiel de conversion et leur niveau d'intérêt pour votre business.

Cette méthode a aboutit à la classification suivante :

- ***CORE*** - '123' - most recent, frequent, revenue generating - core customers that should be considered as most valuable clients.
- ***GONE*** - '311', '312', '313' - gone, one-timers - those clients are probably gone.
- ***ROOKIE*** - '111', '112', '113' - just have joined - new clients that have joined recently.
- ***WHALES*** - '323', '213', '223 - most revenue generating - whales that generate revenue.
- ***LOYAL*** - '221', '222', '321', '322'
- ***REGULAR*** - '121', '122', '211', '212', - average users - just regular customers that don't stand out





	Frequency	Monetary	Recency	R_score	M_score	F_score	RFM_score	segments
0	1	129.90	111	1	3	1	113	Rookies
1	1	18.90	114	1	1	1	111	Rookies
2	1	69.00	537	3	2	1	312	Gone
3	1	25.99	321	3	1	1	311	Gone
4	1	180.00	288	2	3	1	213	Whales
...	...	...	...	...	...	...	...	...
91468	1	1570.00	447	3	3	1	313	Gone
91469	1	64.89	262	2	2	1	212	Regular
91470	1	89.90	568	3	2	1	312	Gone
91471	1	115.00	119	1	2	1	112	Rookies
91472	1	56.99	484	3	1	1	311	Gone

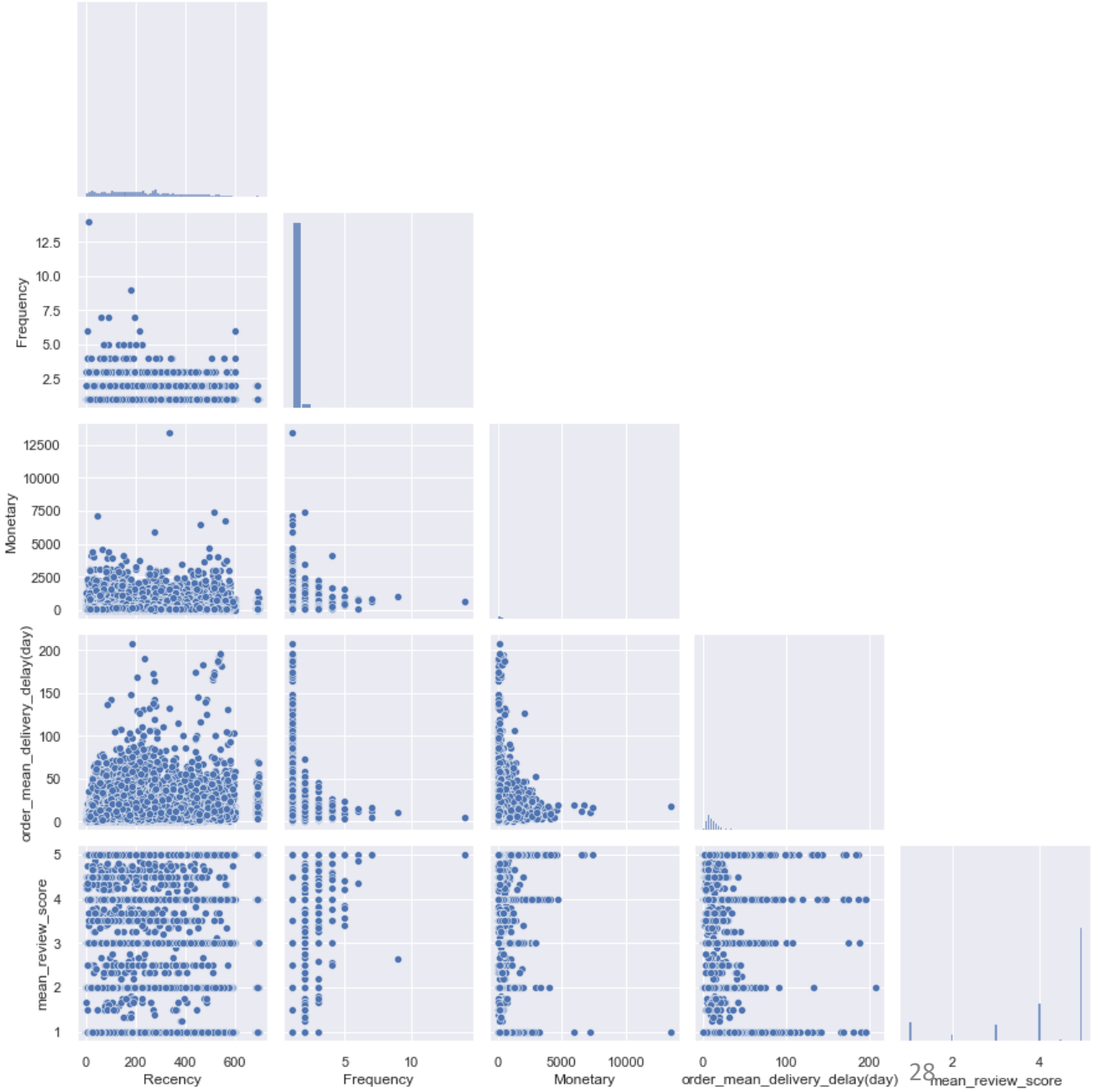
Stratégie comportementales

- Les variables sélectionnées sont :

– Frequency
– Monetary
– Recency
– mean_review_score
– order_mean_delivery_delay(day)

- Aucun groupe clair est détecté sur le pairplot.

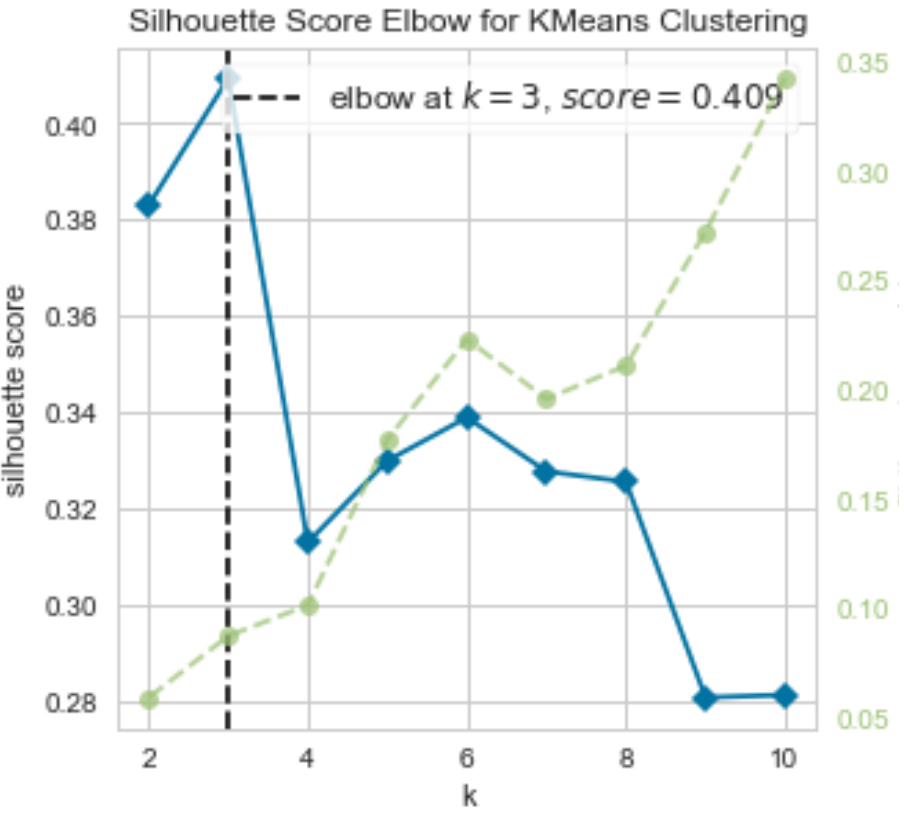
- Le clustering a effectué par 3 méthodes (Kmeans, DBSCAN et hiérarchique. Dans cette partie on va représenter uniquement les résultats du modèle ***Kmeans*** avec réduction dimensionnelle à l'aide des algorithme ***PCA*** et ***t-SNE***.

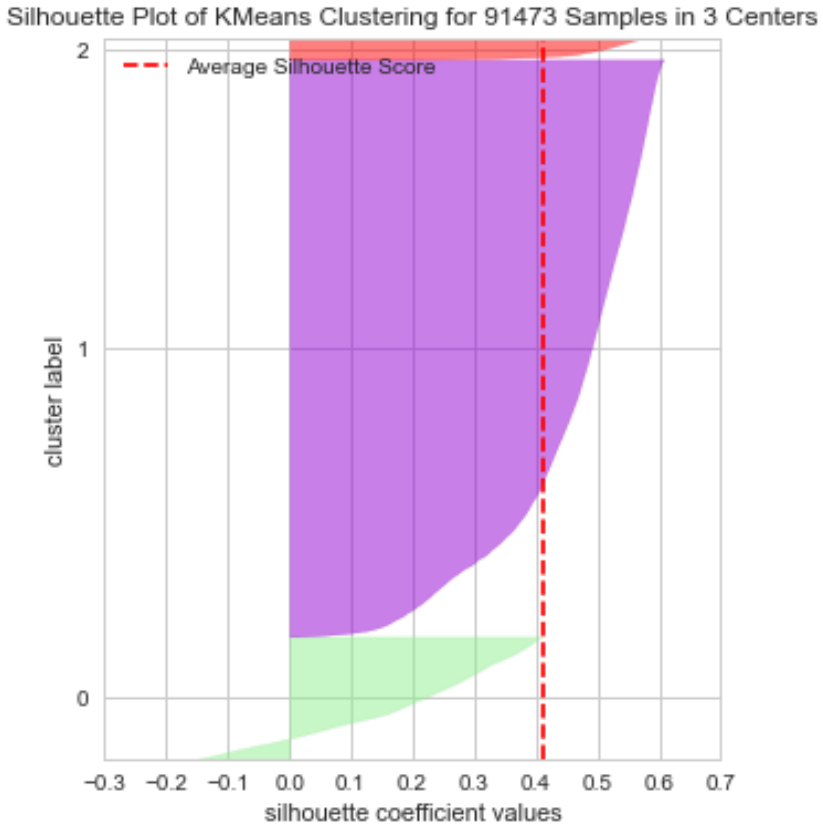


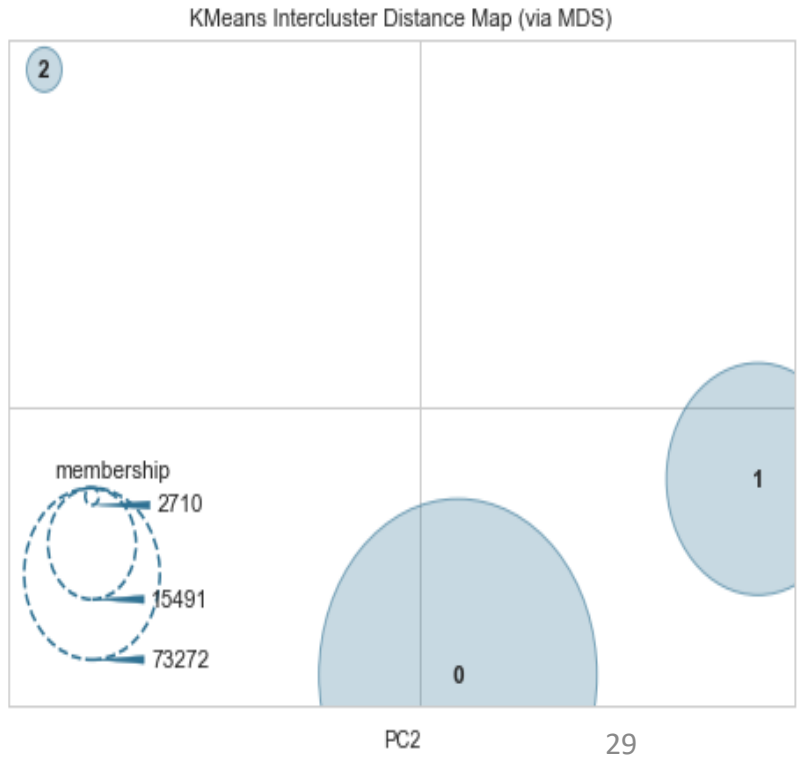
Stratégie comportementales : Kmeans clustering

Optimisation du nombre de clusters k

- K=3 est choisi comme valeur optimale.
- Les groupes sont globalement bien réparties et séparés avec un score de silhouette moyen de **0.409** .
- Le cluster 2 (en rouge) est moins dense que les autres groupes.
- Le cluster 0 montre quelques individus qui sont mal affectés.





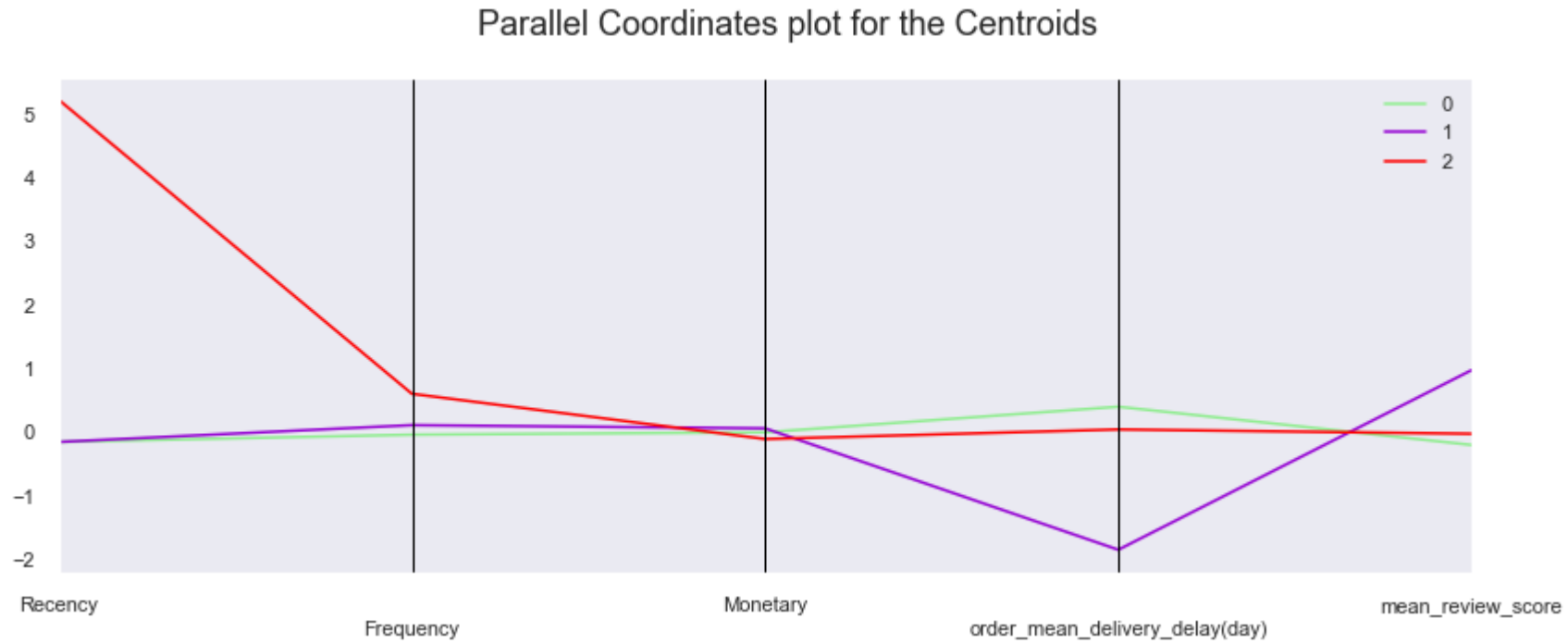


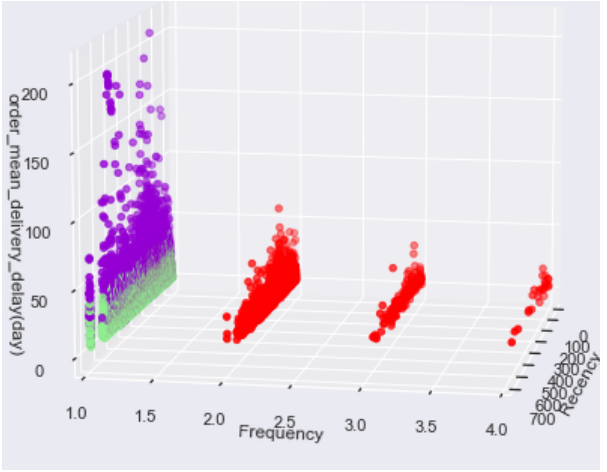
Stratégie comportementales : Kmeans clustering

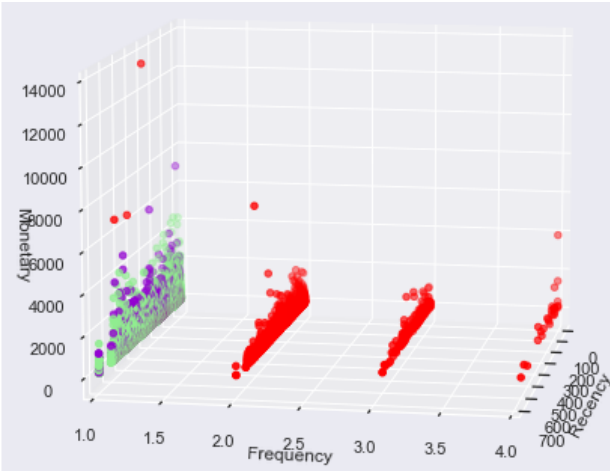
Analyse des clusters

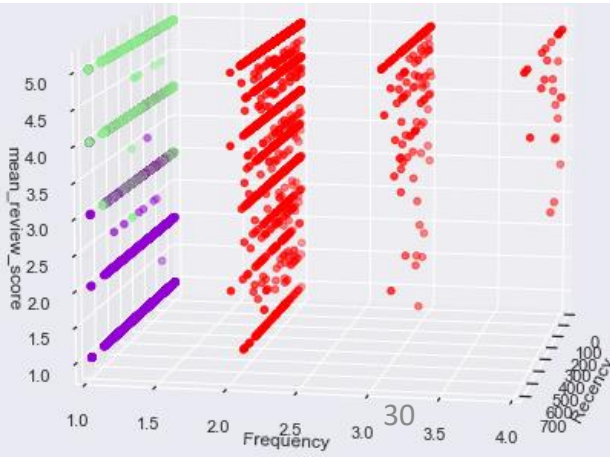
La majorité des clients qui ont passé plus de 2 commandes sont satisfaits par les produits Olist. Les clusters 0 et 1 sont des clients ayant une seule commande avec recency et monetary proches. Cependant, le cluster 1 représente les clients mécontents. Ceci est du au délais de livraison importants.

cluster	Recency	Frequency	Monetary	order_mean_delivery_delay(day)	mean_review_score
0	235.633861	1.000000	131.834376	10.535540	4.658479
1	244.690696	1.000000	164.030821	21.674441	1.777178
2	219.435793	2.109594	269.592756	12.194528	4.203033



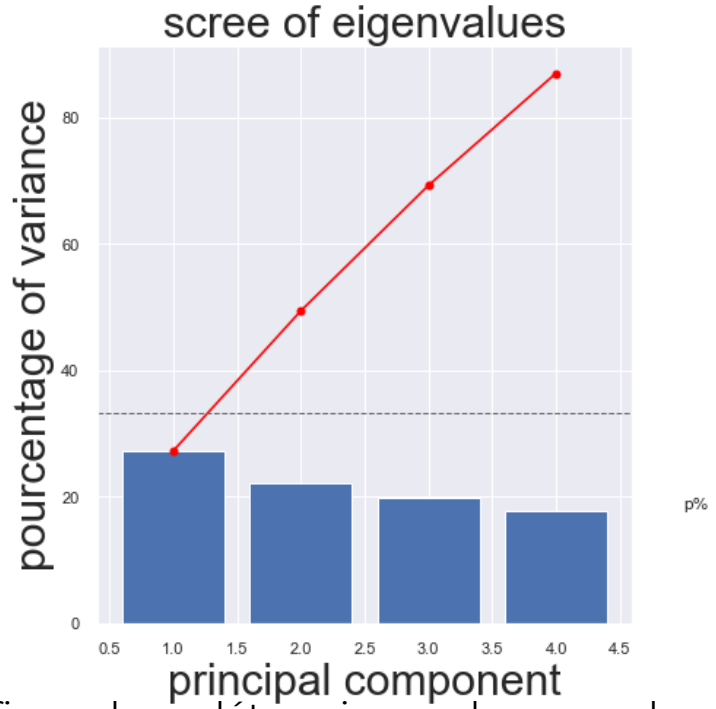




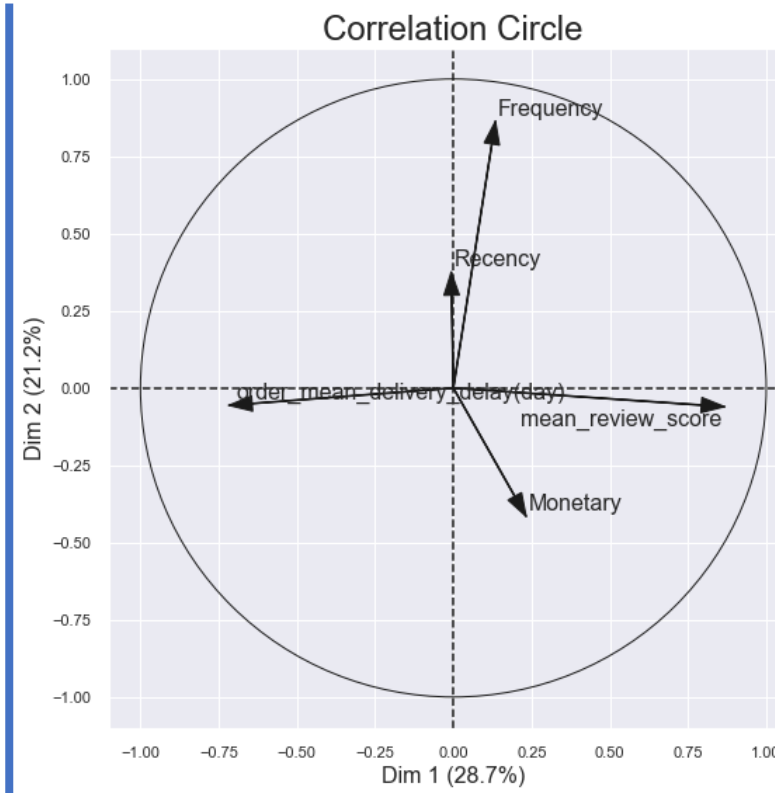


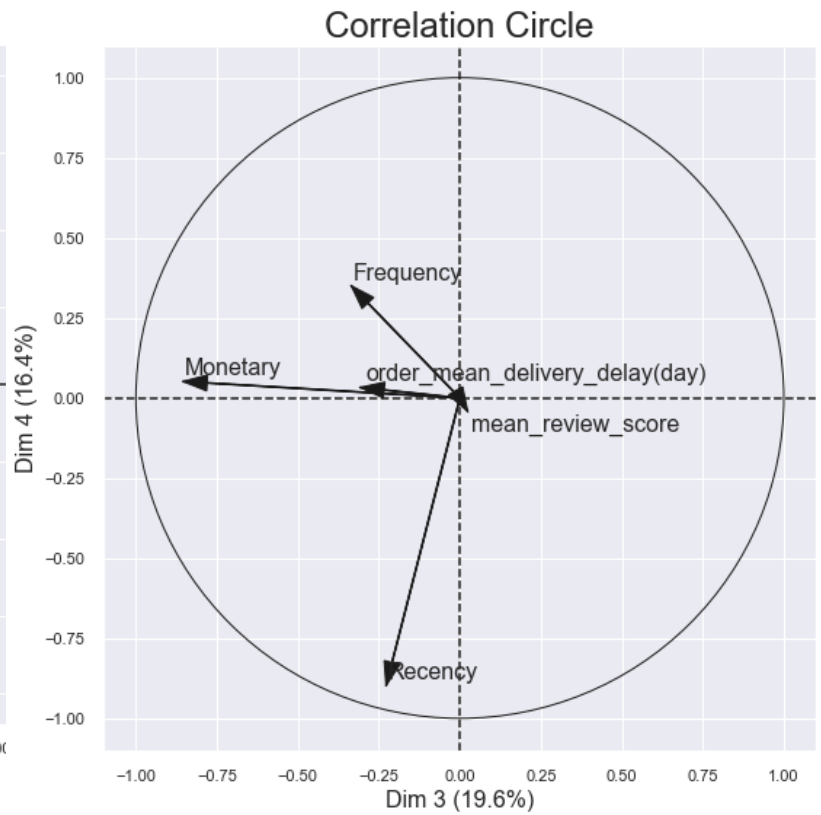
Stratégie comportementales : Kmeans clustering

Réduction dimensionnelle



- Afin de déterminer le nombre de composantes nécessaires à l'analyse, nous projetons les données sur les axes principaux d'inertie qui sont ordonnés selon l'inertie du nuage projeté de la plus grande à la plus petite : c'est l'éboulis des valeurs propres.
- On remarque que les 4 premières composantee couvre plus de 85% d'informations.





- DIM1 représente Le délai de livraison (anticorrélé) & la satisfaction des clients(fortement corrélé).
- Frequency est fortrement corrélée avec DIM2.
- Monetary est fortement corrélée avec DIM3.
- Recency est représentée par DIM4.

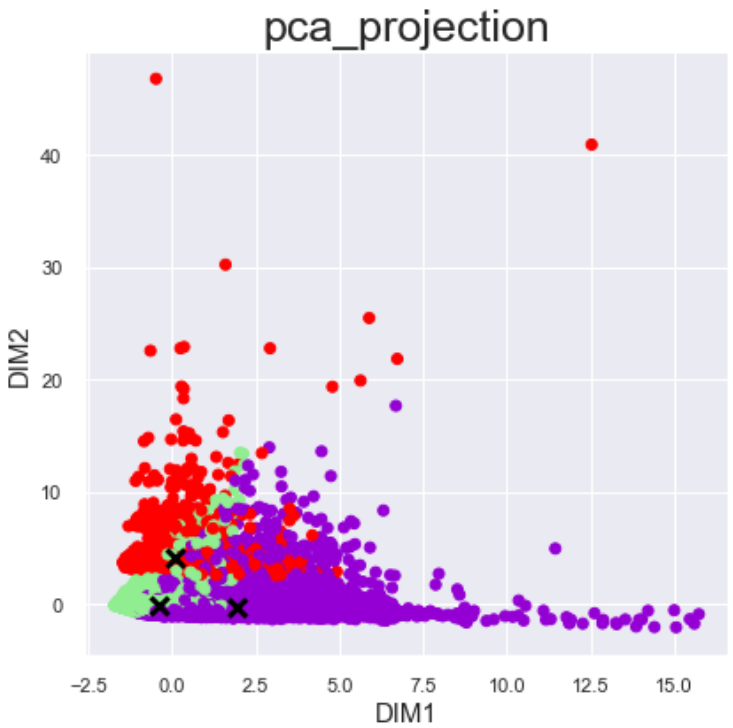
Stratégie comportementales

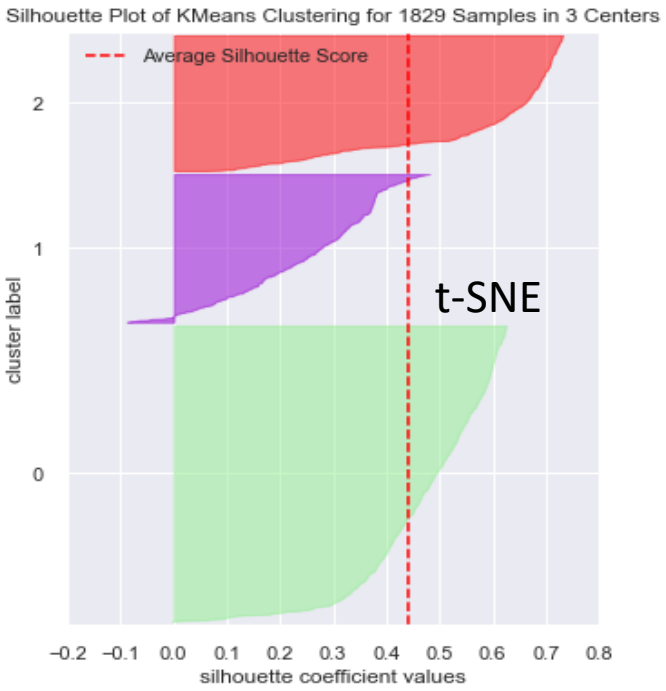
PCA & t-SNE comparaison

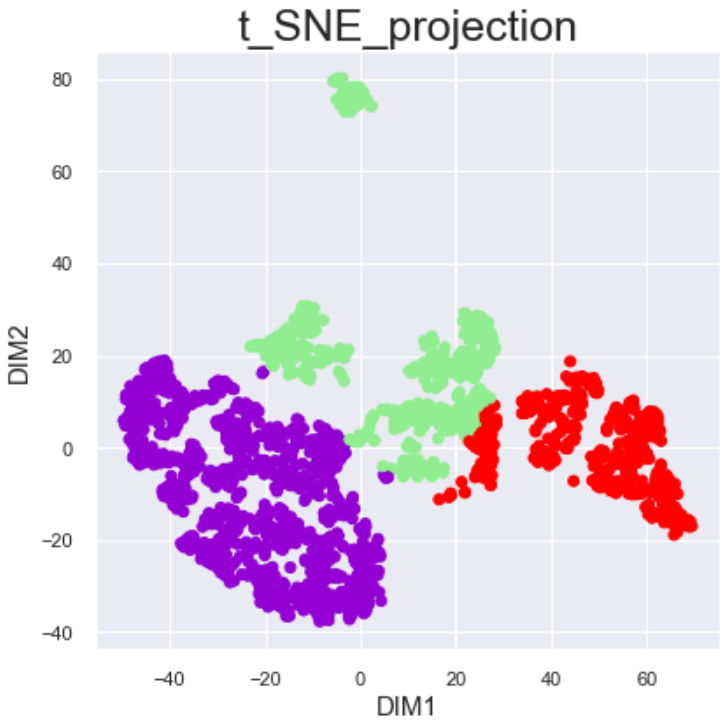
- La réduction dimensionnelle avec PCA n'a pas amélioré la séparation des clusters (même score silhouette avant réduction). Cependant, la méthode t-SNE donne des clusters bien séparés et denses avec une augmentation du score silhouette de 0.409 à 0.44 .

- PCA et t-SNE ont aboutit à 3 clusters avec une densité différentes des groupes.

- Les méthodes de réductions dimensionnelles utilisées permettent des réduire le temps d'entrainement de l'algorithme Kmeans.







Sommaire

1- Problématique et présentation du projet

2- Analyse Exploratoire des Données (EDA)

3- Preprocessing

4- Modeling et optimisation

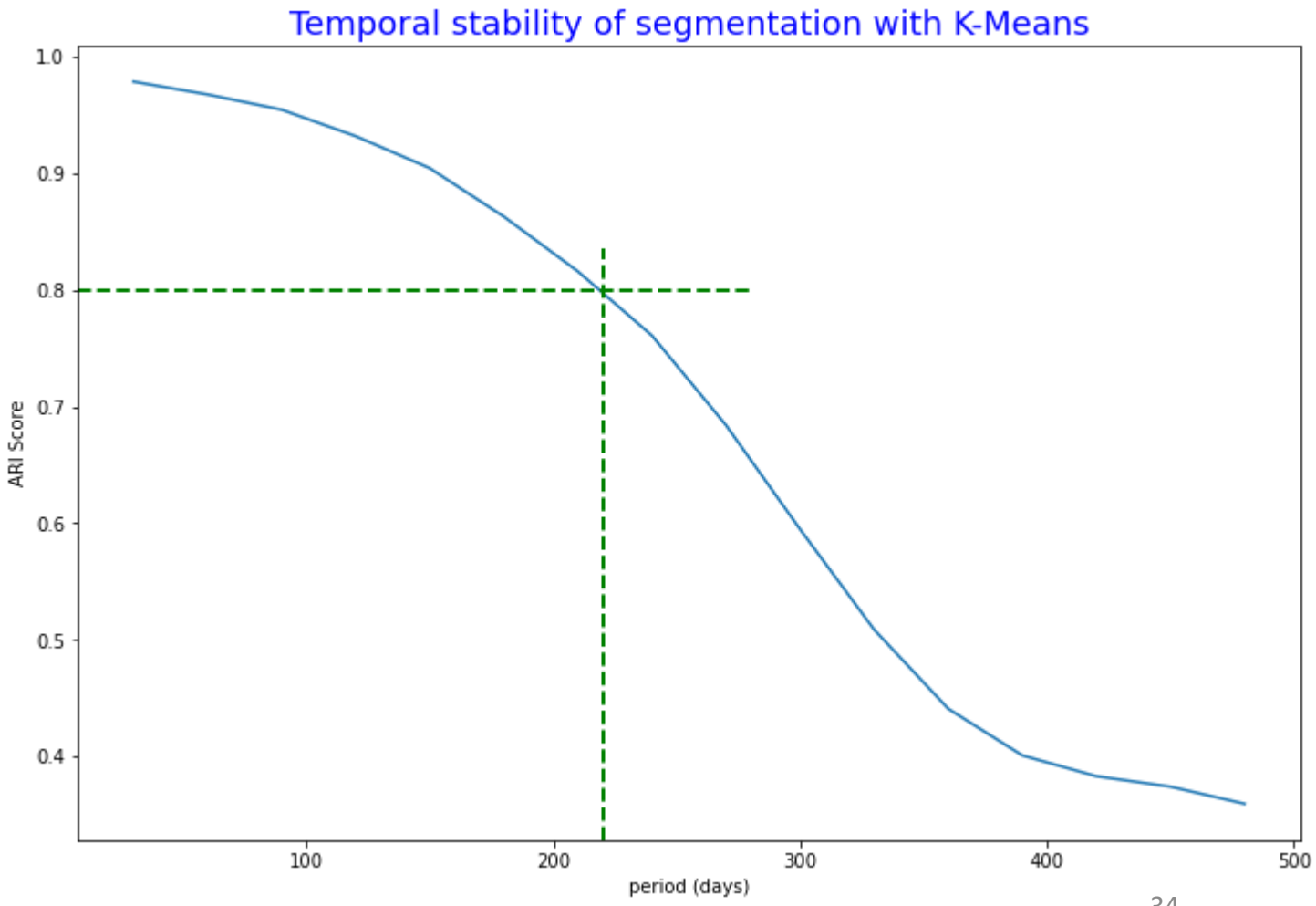
5- Maintenance du modèle

6- Conclusion

Maintenance du modèle

Stabilité temporelle de l'algorithme KMeans

- Dans le but d'établir un contrat de maintenance de l'algorithme de clustering, nous devons testé sa stabilité dans le temps. Le jeu de données d'Olist s'étale du 03/10/2016 jusqu'à 29/08/2018 soit une période de 23 mois. Nous avons fixé une période initiale T0 de 6 mois.
- L'algorithme des Kmeans est itéré sur toute la périodes avec des intervalle de 15 jours (Ti = T0+15n) avec n le nombre d'itérations.
- Le score ARI est calculer pour comparer les labels et vérifier à quel moment les clients changent de segments?
- D'après le graphe, nous avons une inflexion au bout de 220 jours. Il faudra donc prévoir la maintenance du programme de segmentation tous les 220 jours (~7mois).
- À chaque maintenance, il sera nécessaire de redéfinir les clusters.



Sommaire

1- Problématique et présentation du projet

2- Analyse Exploratoire des Données (EDA)

3- Preprocessing

4- Modeling et optimisation

5-Maintenance du modèle

6- Conclusion

Conclusion générale

- Les algorithmes Kmeans et hiérarchique permettent de bien segmenter les clients. La compréhension des différents comportements(satisfaction, montant dépensé, nombre de commandes, ancienneté du client, délai de livraison) aide l'équipe marketing à bien cibler les utilisateurs.

- La réduction dimensionnelle avec l'algorithme t-SNE a donné des clusters bien séparés et denses.

- Au fil du temps, le modèle utiliser pour prédier les clusters n'est plus fiables et ses performances diminue au cours du temps : c'est le data drift. Ceci est du peut être à la collection des données ou le fait que le comportement des clients varie selon la saison,etc. Afin de garder les meilleurs performances de notre modèle, Il faut prévoir une maintenance tous les 220 jours. Cette maintenance peut être faite en changeant le modèle ou en ajoutant d'autres features...

Merci pour votre attention