



Project 5

Segmentez des clients d'un site e-commerce

Tetiana Lemishko

Sommaire:

- Mission et objectifs principaux
- Nettoyage et préparation du jeu de données
- Analyse exploratoire
- Segmentation des client
- Simulation pour déterminer la fréquence nécessaire de mise à jour
- Conclusion

Mission et objectifs principaux

The logo for Olist, featuring the word "olist" in white lowercase letters on a blue rectangular background.

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

Objectifs principaux:

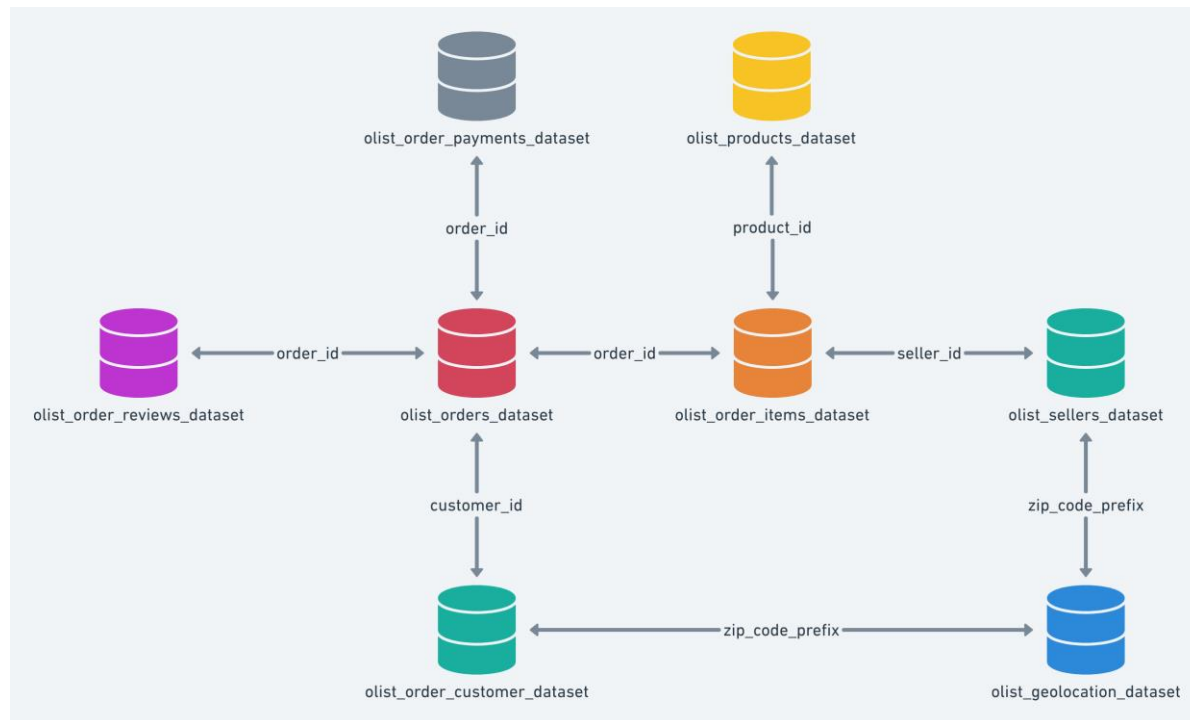
- Comprendre les différents types d'utilisateurs.
- Réaliser une segmentation des clients, facilement exploitable pour l'équipe marketing.
- Evaluer la fréquence à laquelle la segmentation doit être mise à jour.

Jeu de données

Nous avons à notre disposition un jeu de données qui contient des informations sur:

- Les clients
- Les commandes par clients
- Les reviews par commandes
- Les produits
- Les vendeurs
- La géolocalisation des clients et des vendeurs

Il y a plusieurs fichiers CSV, et voici un schéma expliquant comment ils sont liés entre eux :



Merge des données

1) Merge data_customers & data_orders sur la clé customer_id

```
data = pd.merge(data_customers, data_orders, on='customer_id', how='inner')
```

2) Merge data & data_order_reviews sur la clé "order_id"

```
data = pd.merge(data, data_order_reviews, on='order_id', how='inner')
```

3) Merge data & data_order_payments sur la clé order_id

```
data = pd.merge(data, data_order_payments, on='order_id', how='inner')
```

4) Merge data & de data_order_items sur la clé order_id

```
data = pd.merge(data, data_order_items, on='order_id', how='inner')
```

5) Merge data & data_products sur la clé product_id

```
data = pd.merge(data, data_products, on='product_id', how='inner')
```

6) Merge data & data_sellers sur la clé seller_id

```
data = pd.merge(data, data_sellers, on='seller_id', how='inner')
```

7) Merge data & data_product_category sur la clé product_category_name

```
data = pd.merge(data, data_product_category, on='product_category_name', how='inner')
```

115609 lignes
40 colonnes

On ne va pas utiliser les données de géolocalisation. On va utiliser des noms de ville à la place

Nettoyage et préparation du jeu de données

Valeurs manquantes

	Total	Ratio_of_NA(%)	Types
review_comment_title	101808	88.06	object
review_comment_message	66703	57.70	object
order_delivered_customer_date	2400	2.08	object
order_delivered_carrier_date	1195	1.03	object
order_approved_at	14	0.01	object
product_weight_g	1	0.00	float64
product_width_cm	1	0.00	float64
product_height_cm	1	0.00	float64
product_length_cm	1	0.00	float64
shipping_limit_date	0	0.00	object
price	0	0.00	float64
freight_value	0	0.00	float64
product_category_name	0	0.00	object
product_name_lenght	0	0.00	float64
customer_id	0	0.00	object
product_description_lenght	0	0.00	float64
product_photos_qty	0	0.00	float64
product_id	0	0.00	object
seller_zip_code_prefix	0	0.00	int64
seller_city	0	0.00	object

Suppression des colonnes inutiles

'customer_id'
'review_comment_title'
'review_creation_date'
'review_answer_timestamp'
'review_comment_message'
'product_category_name'
'product_name_lenght'
'product_description_lenght'
'product_weight_g'
'product_width_cm'
'product_length_cm'
'product_height_cm'
'seller_zip_code_prefix'

On a supprimé les valeurs manquantes restantes pour les variables 'order_approved_at', 'order_delivered_carrier_date', 'order_delivered_customer_date' (ce qui n'est pas une grosse perte d'informations car ces valeurs manquantes ne représentent qu'un petit pourcentage des données)

Nettoyage et préparation du jeu de données

Feature Engineering

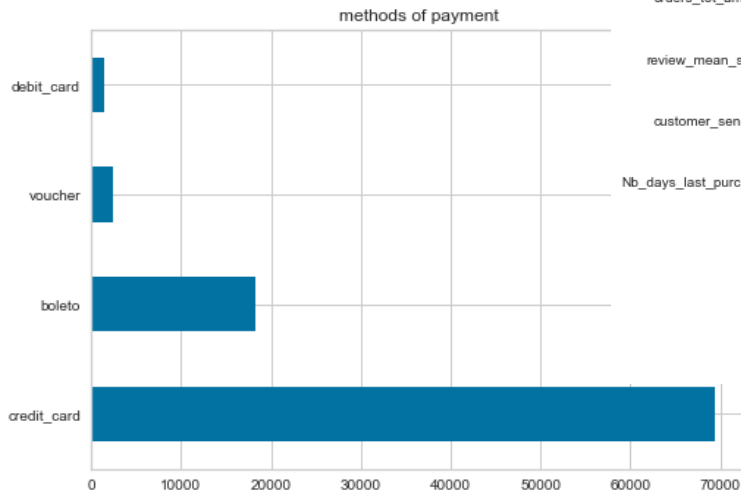
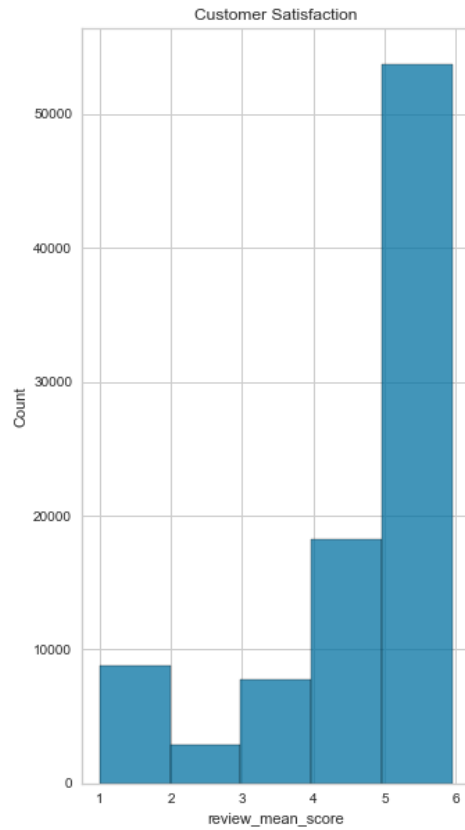
On a calculé plusieurs nouvelles variables que l'on va utiliser pour le clustering

- Montant total des achats par client ('orders_tot_amount')
- Note moyenne des commentaires par client ('review_mean_score')
- Le nombre de jours depuis la dernière commande par client ('Nb_days_last_purchase')
- Le nombre de commandes par client ('nb_order')

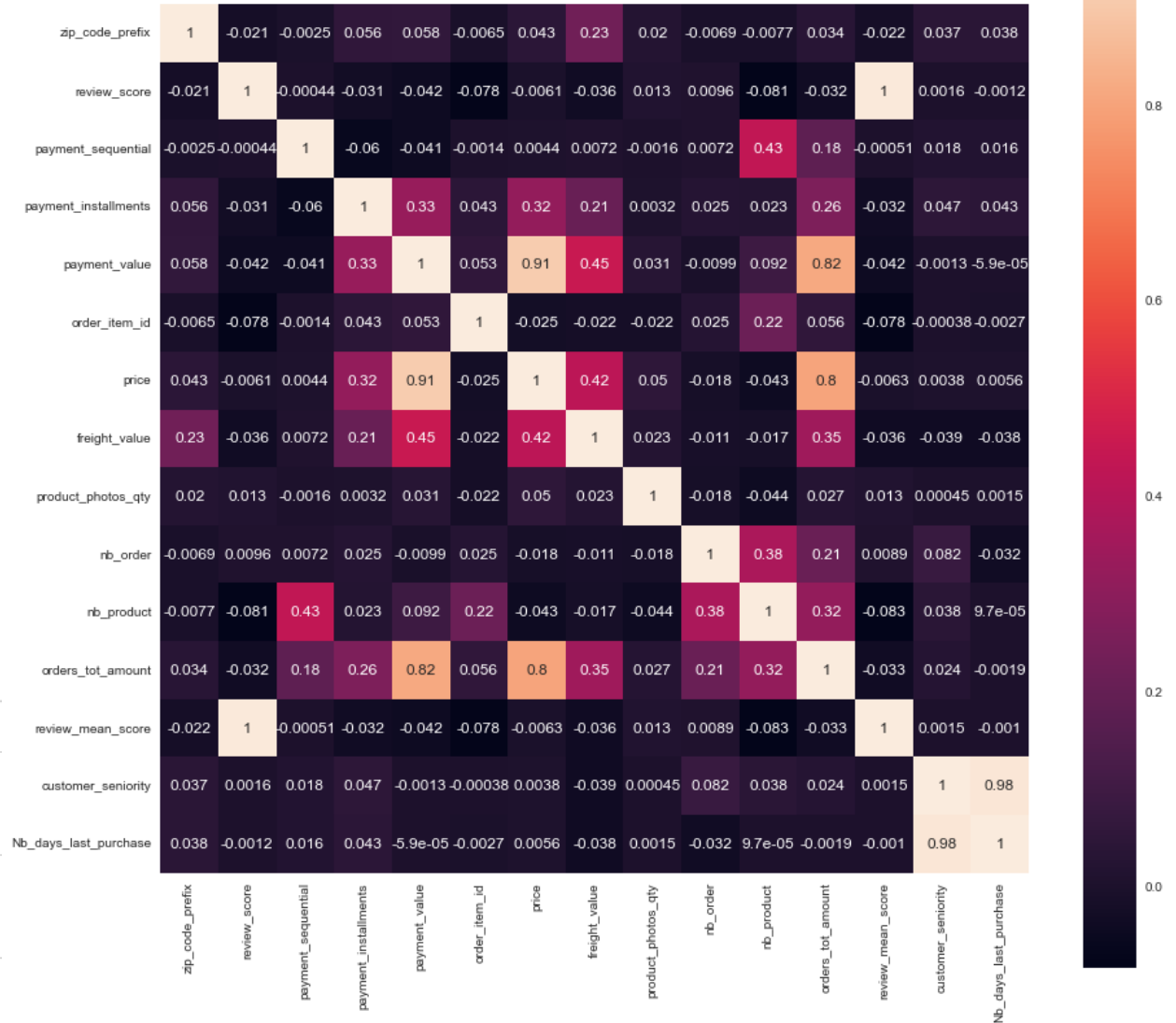
Traitement des doublons

```
data = data.drop_duplicates('customer_unique_id')
```

Analyse exploratoire des données



Corrélations entre variables



Clustering

Choix des features

- Montant total des achats par client
- Note moyenne des commentaires par client
- Le nombre de jours depuis la dernière commande par client
- Nombre de commandes par client

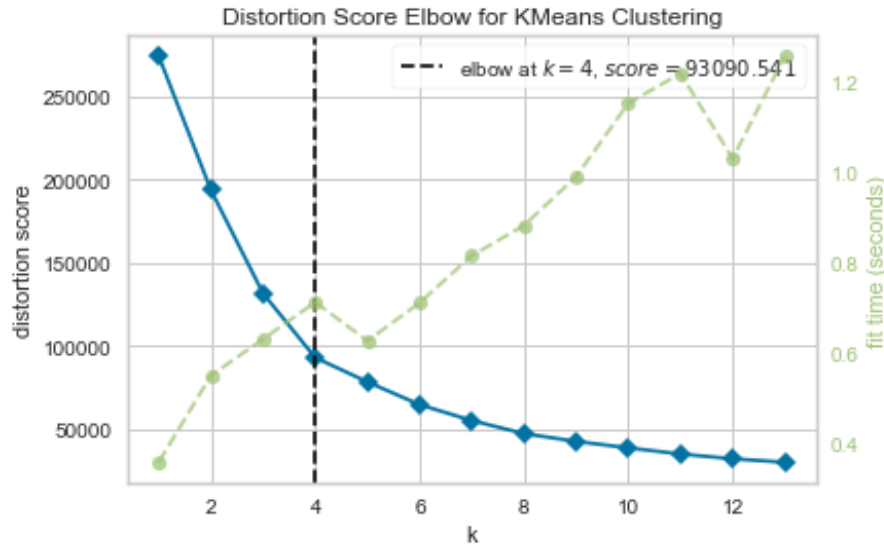
Choix de l'algorithme de Clustering

- K-Means
- AgglomerativeClustering (hierarchical clustering)
- DBSCAN

K-Means clustering (3 variables)

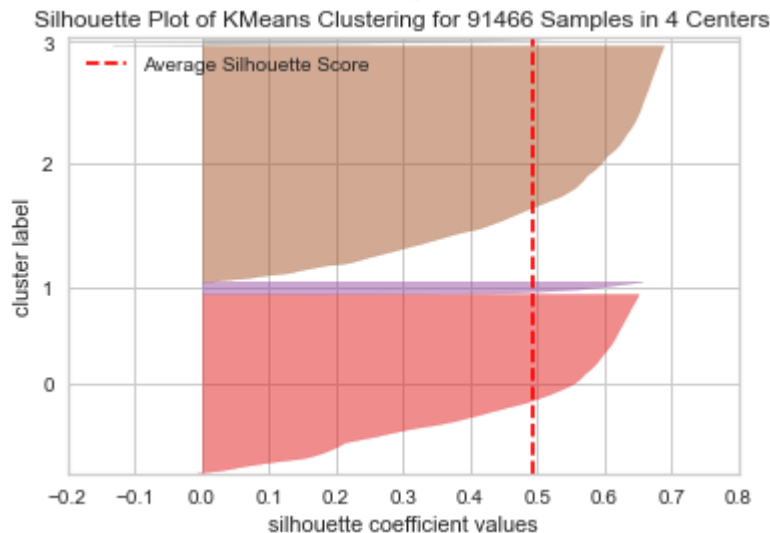
- Montant total des achats par client
- Le nombre de jours depuis la dernière commande par client
- Nombre de commandes par client

Pour déterminer le nombre de clusters (la valeur optimale de k), on va utiliser la méthode du coude et le silhouette score



Méthode du coude:

La valeur retenue pour k est celle qui marque le début d'un pallier : pour des valeurs inférieures la qualité de regroupement est nettement moins bonne, alors que pour des valeurs supérieures la qualité ne s'améliore pas sensiblement.



Silhouette Score :

Pour chaque point, son coefficient de silhouette est la différence entre la distance moyenne avec les points du même groupe que lui et la distance moyenne avec les points des autres groupes voisins.

K-Means clustering (3 variables)

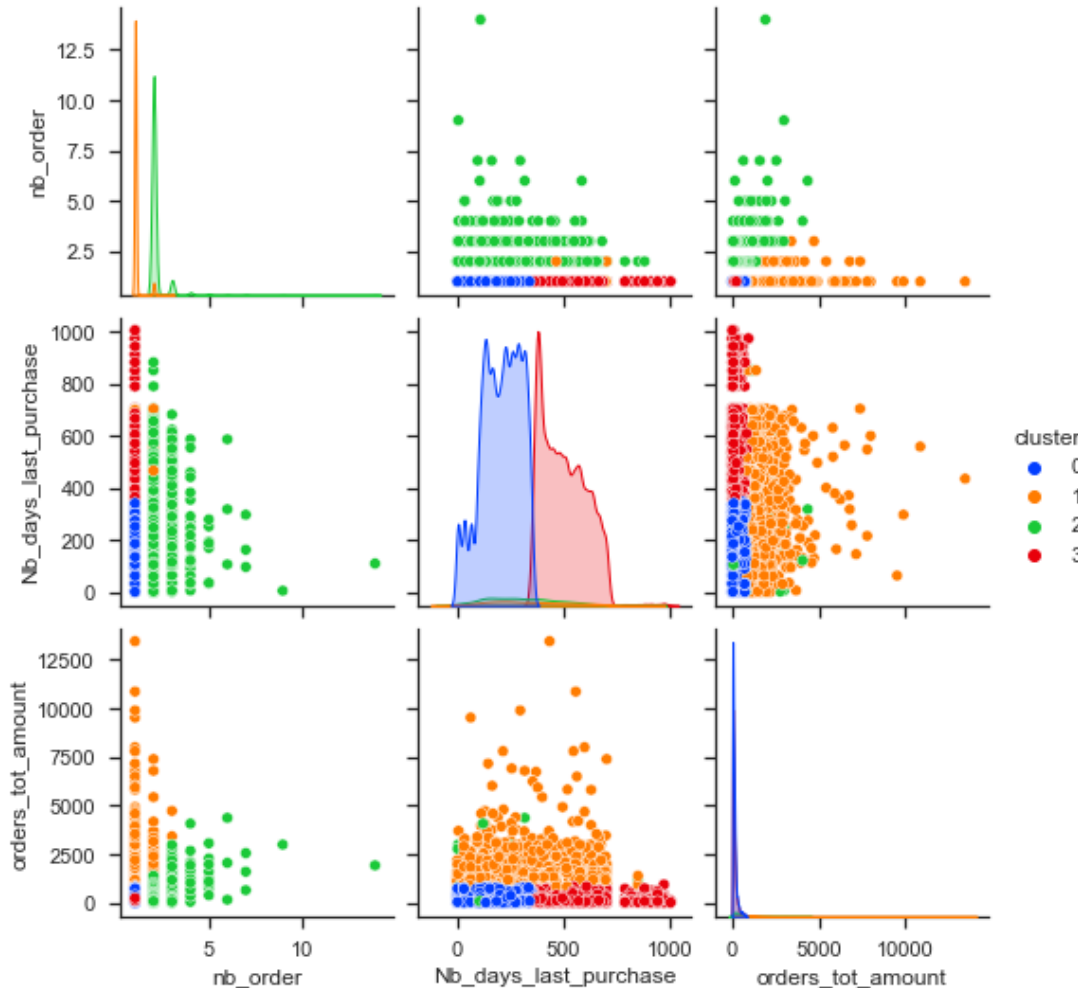
Caractérisation des clusters

Cluster 0: Les clients qui ont commandé récemment et qui ne commandent pas et ne dépensent pas beaucoup

Cluster 1: Les clients qui ont commandé récemment et il y a longtemps et qui dépensent beaucoup mais avec le nombre des commandes bas

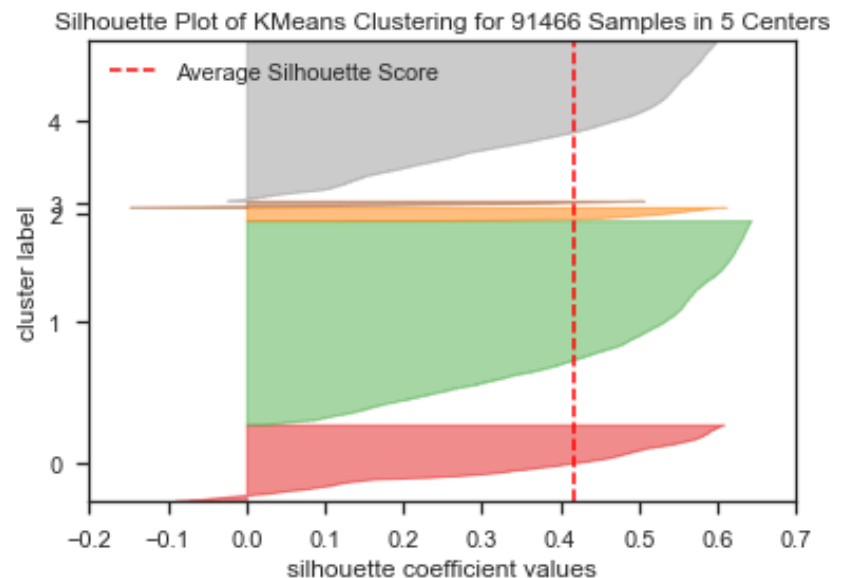
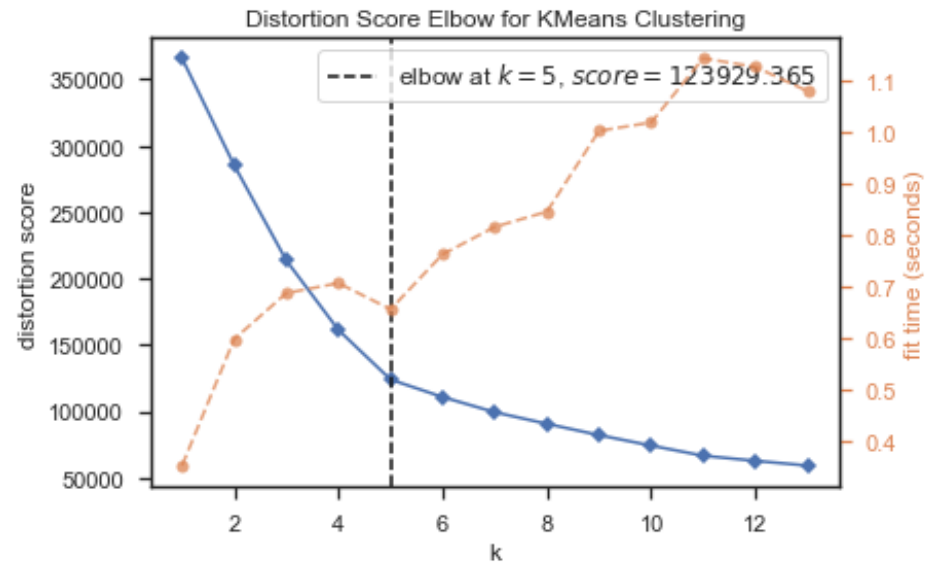
Cluster 2: Les clients qui ont commandé récemment et qui dépensent un montant moyen avec beaucoup de commandes

Cluster 3: Les clients qui n'ont pas commandé depuis longtemps et qui ne dépensent pas et ne commandent pas beaucoup



K-Means clustering (4 variables)

- Montant total des achats par client
- Note moyenne des commentaires par client
- Le nombre de jours depuis la dernière commande par client
- Nombre de commandes par client



K-Means clustering (4 variables)

Caractérisation des clusters

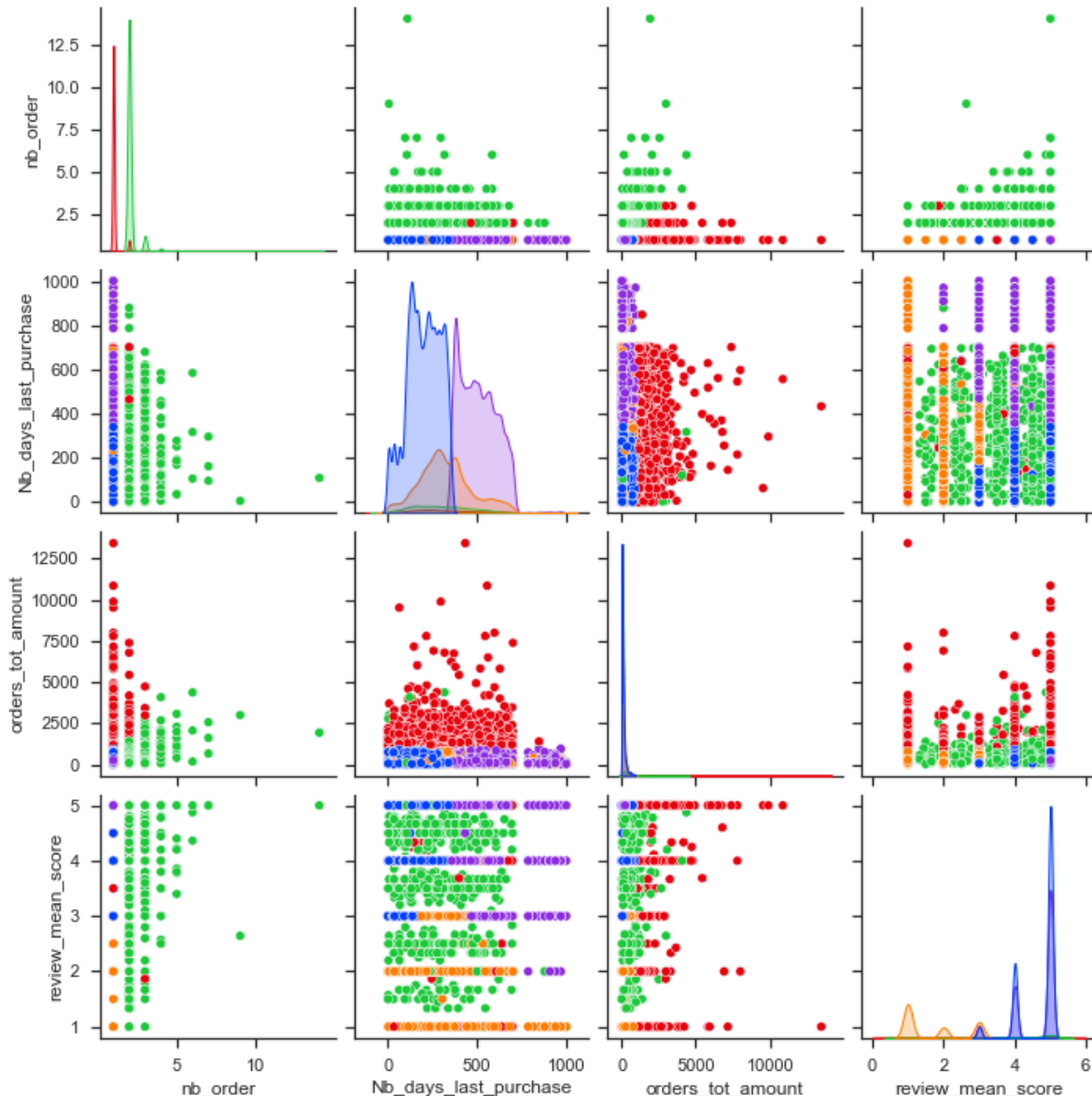
Cluster 0: Les clients qui ont commandé récemment, qui ne dépensent pas beaucoup, qui ne commandent pas beaucoup et donnent des notes de révision plutôt positives

Cluster 1: Les clients qui n'ont pas commandé depuis longtemps, qui ne dépensent pas beaucoup, qui ne commandent pas beaucoup et donnent des notes de révision plutôt négatives

Cluster 2: Les clients qui ont commandé récemment, qui dépensent la montant moyenne avec beaucoup de commandes et donnent des notes de révision plutôt positives

Cluster 3: Les clients qui ont commandé récemment et il y a longtemps, qui dépensent et commandent beaucoup et donnent des notes de révision plutôt positives

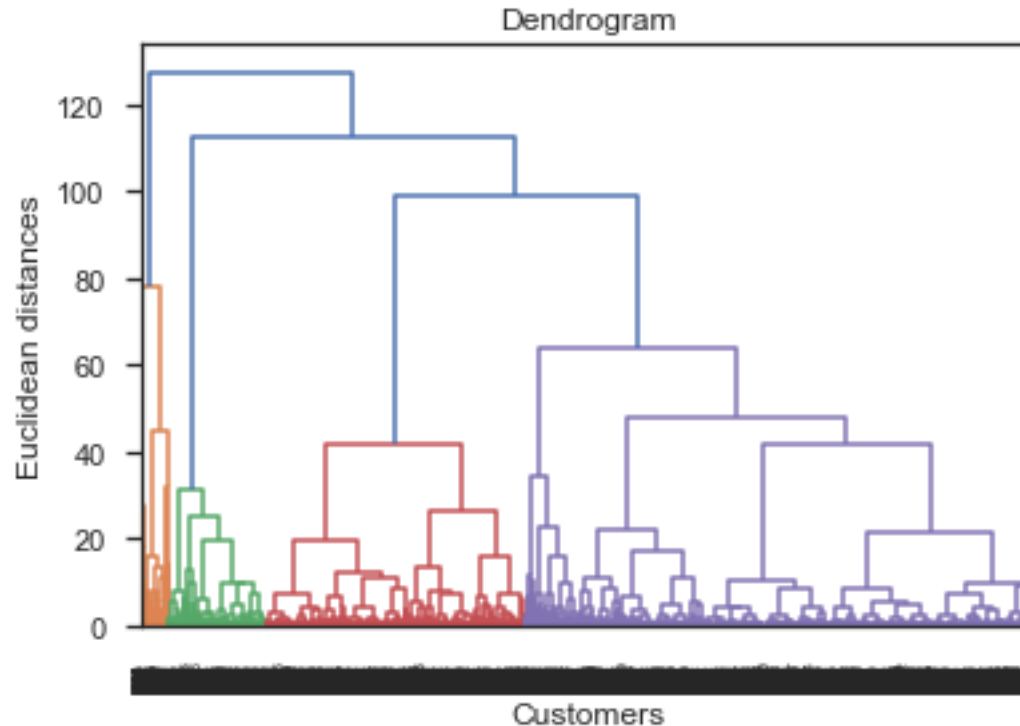
Cluster 4: Les clients qui n'ont pas commandé depuis longtemps, qui ne dépensent pas beaucoup, qui ne commandent pas beaucoup et donnent des notes de révision plutôt positives



Hierarchical clustering (4 variables)

Le volume de données est trop grande pour cet algorithme.

La solution est de considérer un échantillon de données (10%)



Le nombre optimal de clusters est 5

Hierarchical clustering (4 variables)

Caractérisation des clusters

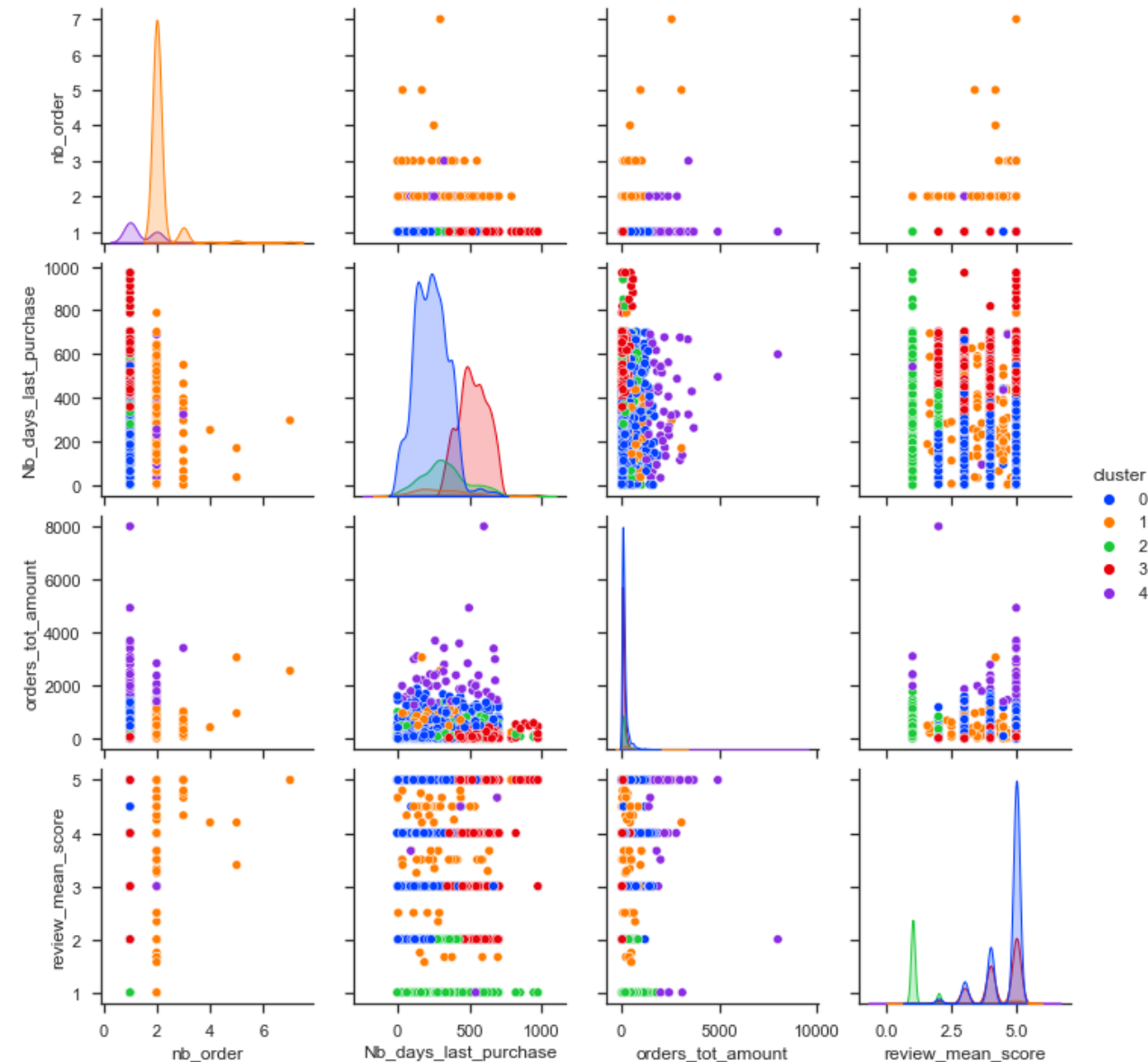
Cluster 0: Les clients qui ont commandé récemment et il y a longtemps, qui ne dépensent pas et ne commandent pas beaucoup et donnent des notes de révision plutôt positives

Cluster 1: Les clients qui ont commandé récemment, qui dépensent la montant moyenne avec beaucoup de commandes et donnent des notes de révision plutôt positives

Cluster 2: Les clients qui ont commandé récemment et il y a longtemps, qui ne dépensent pas beaucoup, qui ne commandent pas beaucoup et donnent des notes de révision plutôt négatives

Cluster 3: Les clients qui n'ont pas commandé depuis longtemps, qui ne dépensent pas beaucoup, qui ne commandent pas beaucoup et donnent des notes de révision plutôt positives

Cluster 4: Les clients qui ont commandé récemment et il y a longtemps, qui dépensent et commandent beaucoup et donnent des notes de révision plutôt positives

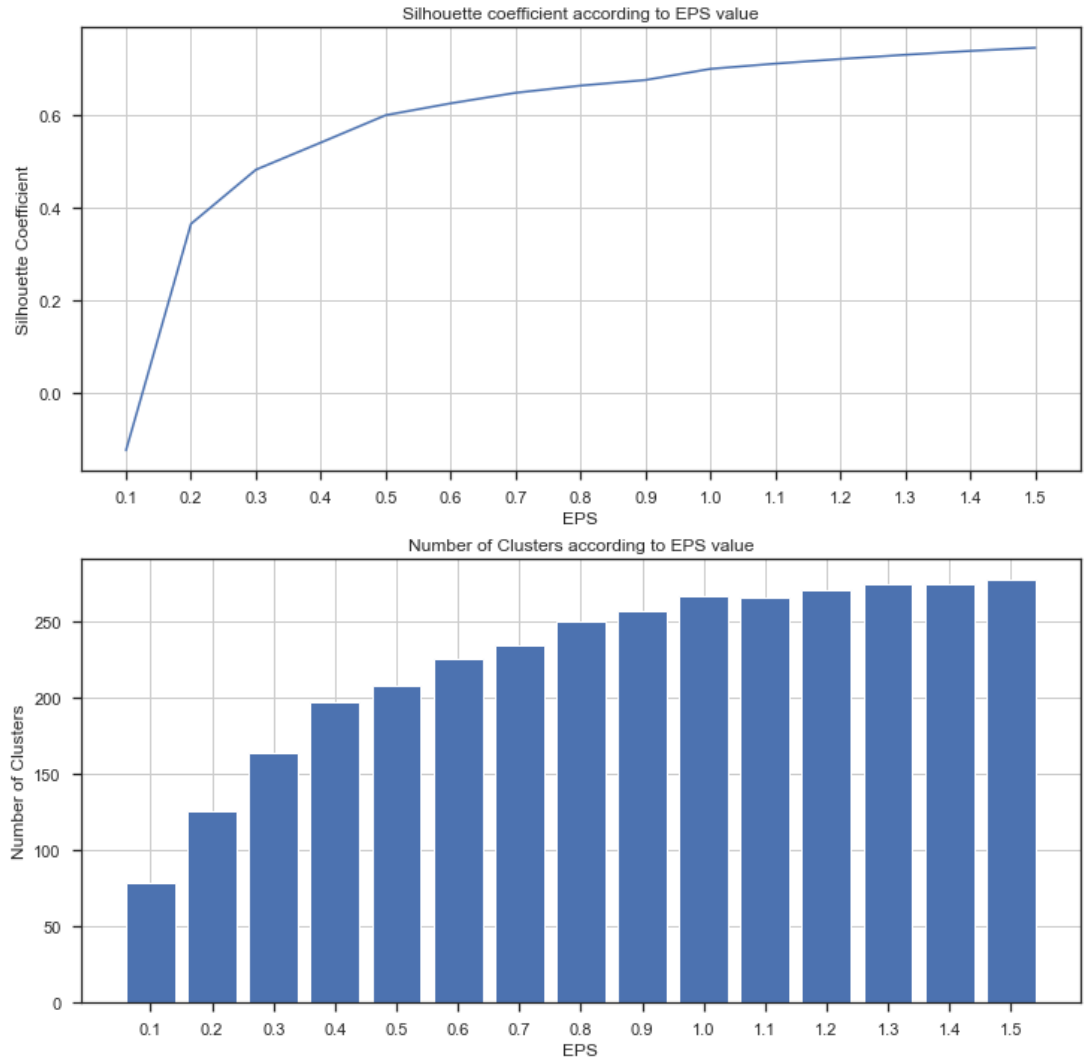


DBSCAN

2 paramètres :





- La distance epsilon
- Le nombre minimum de points MinPts devant se trouver dans un rayon epsilon pour que ces points soient considérés comme un cluster

L'utilisation de DBSCAN dans notre cas est inexploitable pour une utilisation métier car le nombre de clusters est élevé et les clusters ne sont pas équilibrés.



Clusterisation

Choix de l'algorithme de Clusterisation

- K-Means 
- AgglomerativeClustering (hierarchical clustering)  
- DBSCAN 

On préférerait le modèle k-means au clustering hiérarchique car en cas de clustering hiérarchique, seule une fraction des données est utilisée. Cela signifie que cet algorithme pourrait perdre la précision par rapport à k-means

Performance du modèle au cours du temps

On va évaluer la performance de notre modèle (k-means) au cours du temps.

L'objectif est de trouver à partir de qu'elle fréquence le modèle se dégrade, c'est à dire un seuil à partir duquel les prédictions entre le modèle d'origine et un nouveau modèle entraîné sont trop différentes.

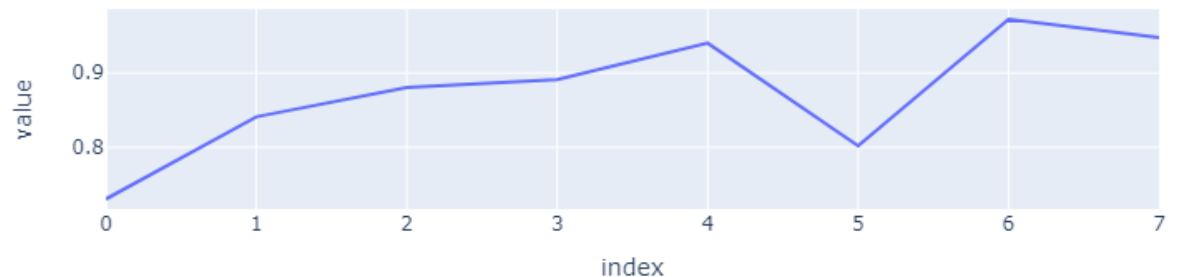
Score ARI par interval de 7D jours (moyenne = 0.9609541003681886)



Score ARI par interval de 14D jours (moyenne = 0.9326281743932807)

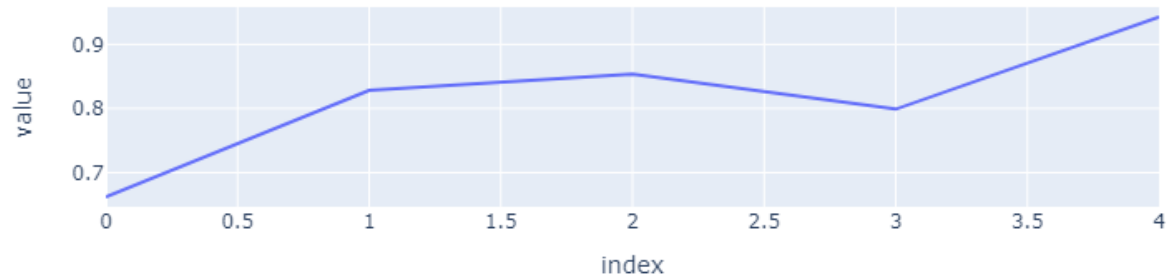


Score ARI par interval de 30D jours (moyenne = 0.8755499777986112)

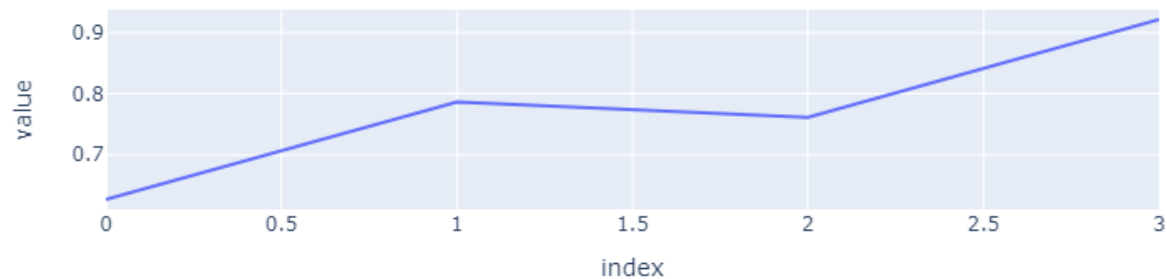


Performance du modèle au cours du temps

Score ARI par interval de 45D jours (moyenne = 0.8170985502141519)



Score ARI par interval de 60D jours (moyenne = 0.7741691537734715)



Score ARI par interval de 90D jours (moyenne = 0.6278401921993523)



Si l'on fixe le seuil du score **ARI à 0.8**, il faudrait donc réentraîner le modèle tous les **45 jours** environ en y ajoutant les nouvelles données.

Conclusions:

- On a choisi 4 caractéristiques pour le clustering : montant total dépensé par client, nombre de jours depuis la dernière commande de chaque client, satisfaction de client et nombre de commandes par client.
- K-means avec 5 clusters a été choisi comme l'algorithme optimal pour le clustering. On a défini certaines caractéristiques du client pour chaque cluster.
- On a trouvé qu'il faut donc ré entraîner le modèle tous les 45 jours environ en y ajoutant les nouvelles données.