# Projet numéro 5 :

## Segmentez des clients d'un site e-commerce

olist

Thomas Zilliox

Février 2025

# Présentation du Projet

olist

## Contexte

- **Construire et maintenir le dashboard** au service des équipes Customer Experience de Olist
- **Réaliser des segmentations des clients** qu'elles pourront utiliser au quotidien pour leurs campagnes de communication.

## Mission

- **Fournir une description actionnable** de la segmentation créée et de sa logique sous-jacente
- **Proposer un contrat de maintenance** basée sur une analyse de la stabilité des segments au cours du temps

# Sommaire

Résumé des étapes du projet :

1) Scripts SQL

2) Préparation des données et analyses exploratoires

3) Algorithmes de classification et visualisations des résultats

4) Contrat de maintenance

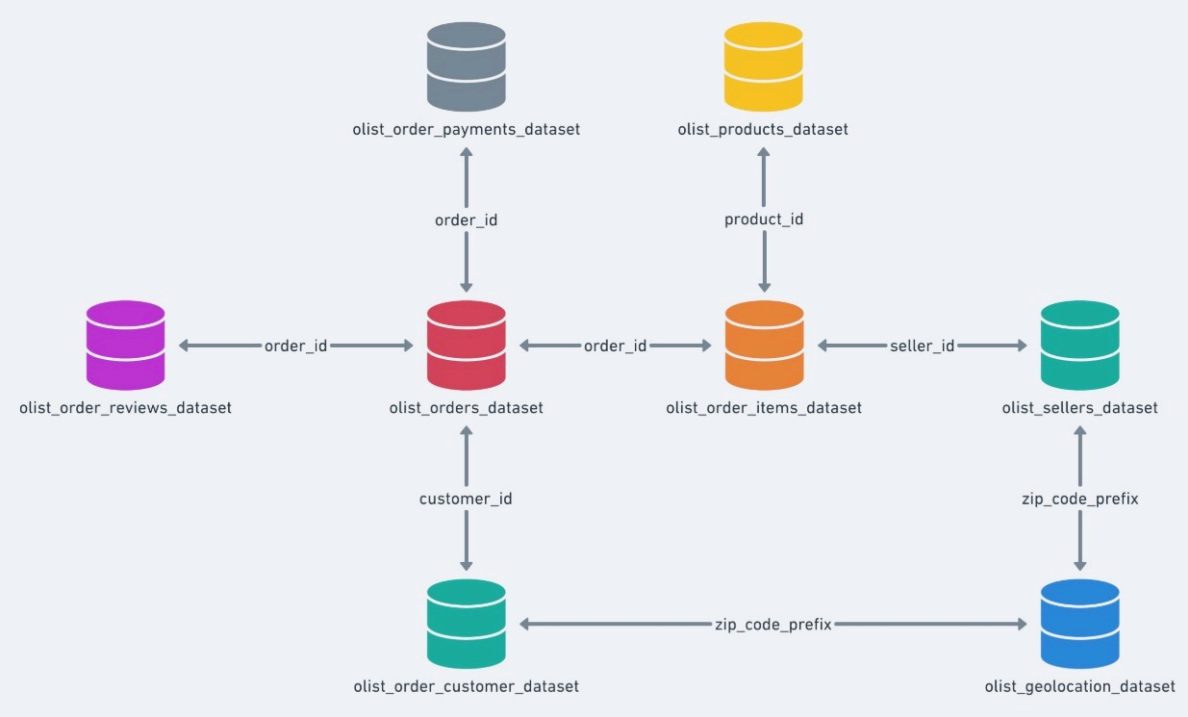
# **Première étape :** Scripts SQL

## **Quatre questions :**

- En excluant les commandes annulées, quelles sont les commandes récentes de moins de 3 mois que les clients ont reçues avec au moins 3 jours de retard ?
- Qui sont les vendeurs ayant généré un chiffre d'affaires de plus de 100 000 Real sur des commandes livrées via Olist ?
- Qui sont les nouveaux vendeurs (moins de 3 mois d'ancienneté) qui sont déjà très engagés avec la plateforme (ayant déjà vendu plus de 30 produits) ?
- Question : Quels sont les 5 codes postaux, enregistrant plus de 30 reviews, avec le pire review score moyen sur les 12 derniers mois ?

# **Première étape :** Scripts SQL

- 9 Bases de données à rassembler en une seule :
- Données concernées : Clients, vendeurs, commandes, produits commandés et reviews




+ product_category_name
Pour la traduction de l'espagnol
À l'anglais

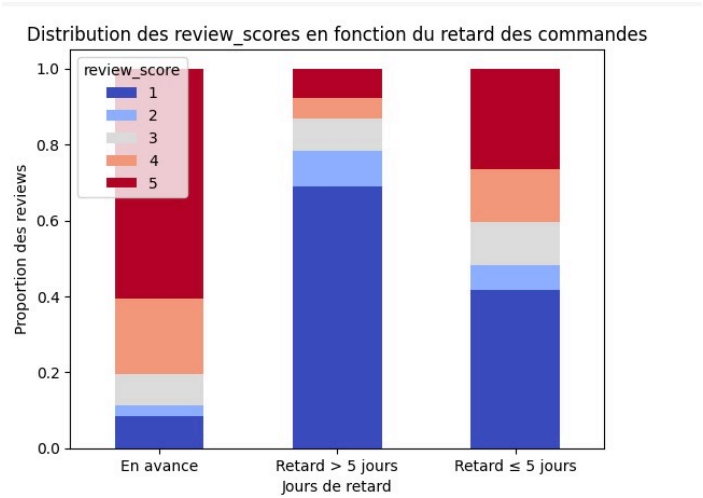
# Deuxième étape : Préparation et première analyse exploratoire des données

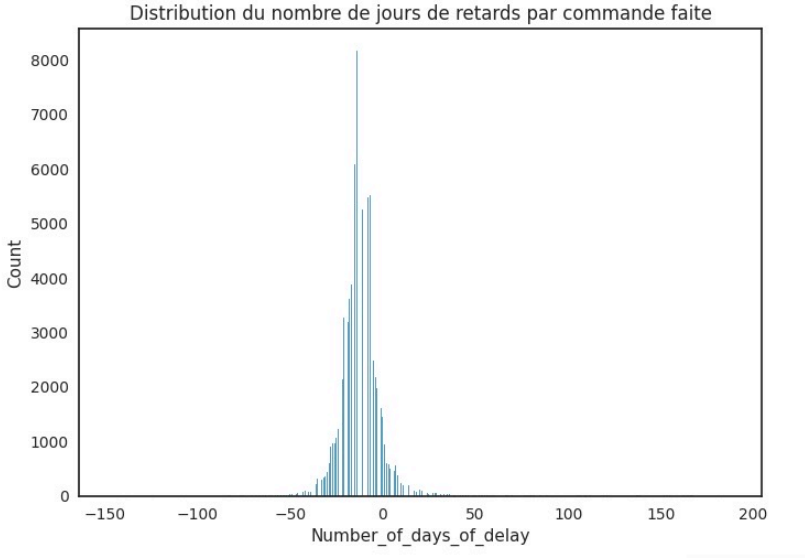
**Deux parties :**

- Premières analyses simples et création des variables RFM (Récence/Fréquence/Montant)

- Visualisations des différents groupes de clients trouvés

# Deuxième étape : Préparation et première analyse exploratoire des données



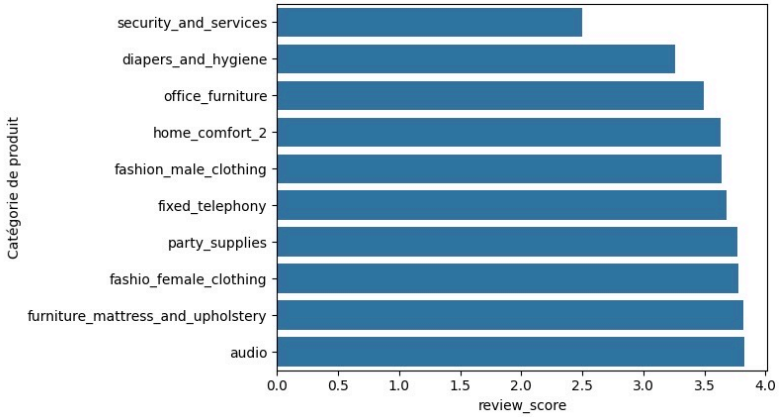




- Majoritairement, les commandes arrivent en avance.
- Plus elles arrivent en retard, plus le review_score sera bas.

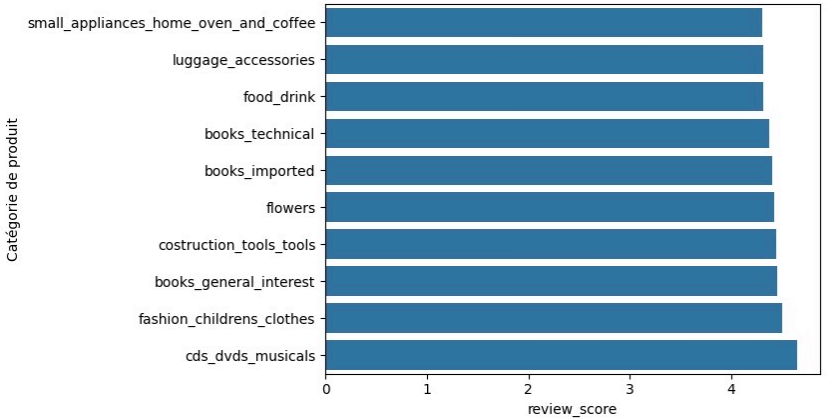
# Deuxième étape : Préparation et première analyse exploratoire des données

Review score des catégories de produits :

- Les 10 produits les moins bien notés

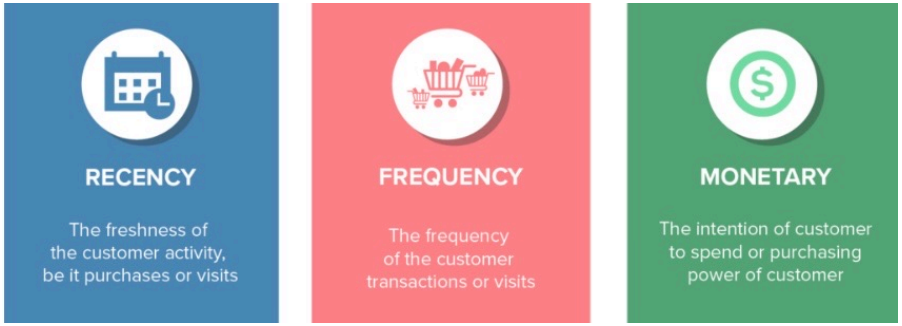


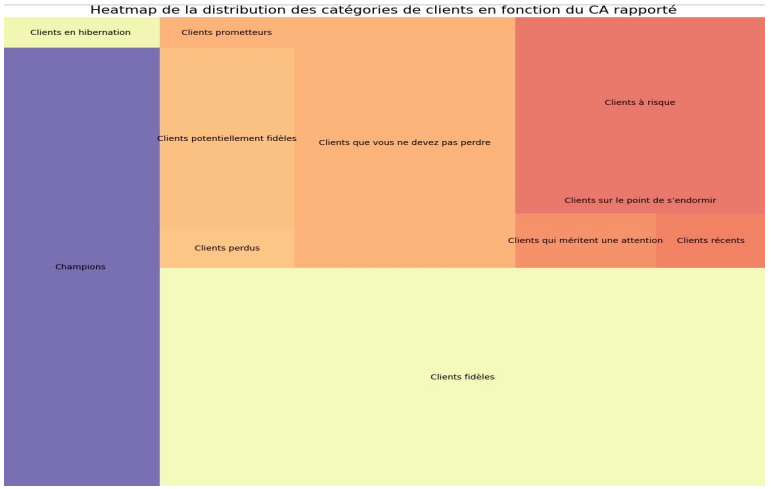

- Les 10 produits les mieux notés



# Deuxième étape : Préparation et première analyse exploratoire des données

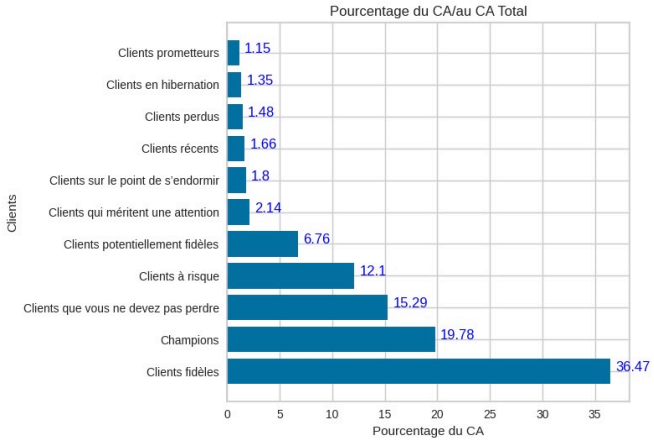
**Première segmentation RFM :**




**11 catégories de clients basés sur deux indicateurs :**

- **Récence du dernier achat**
- **Fréquence et Montant des achats**



# Troisième étape : Algorithme de classification

- Premières phases de test avec Kmeans simple, OneHotEncoding et Standardisation des variables :

- Deuxièmes phases de test avec algorithmes plus complexes :
  - DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
  - CAH (Classification Ascendante Hiérarchique)

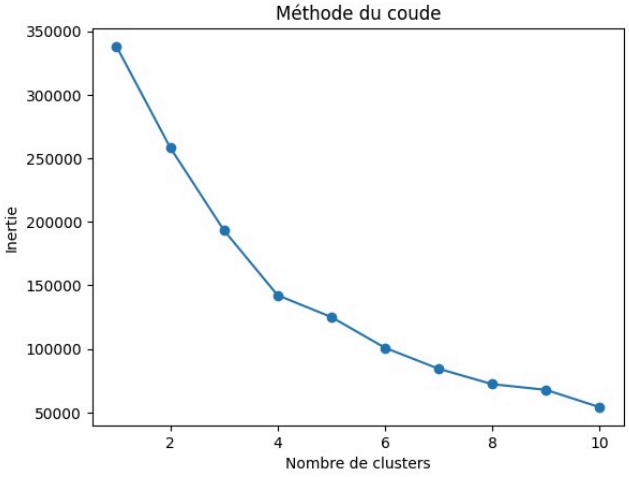
- Types de visualisations réalisés :
  - Nuages de points
  - Radar plots

# Troisième étape : Algorithme de classification

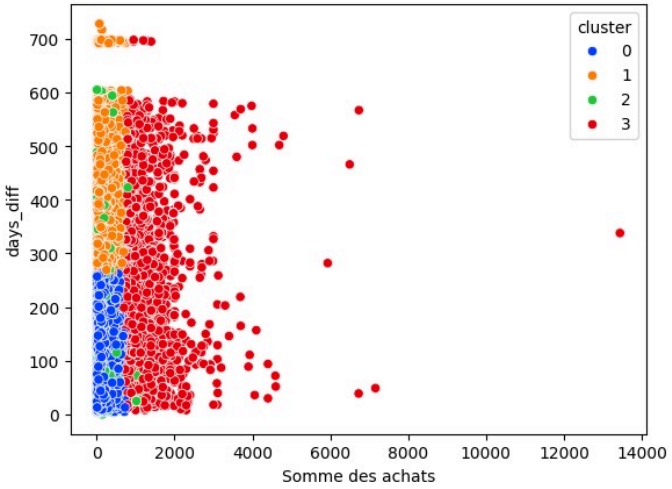
Premières phases de test avec Kmeans simple :

Variables utilisées : ['Somme des achats', 'days_diff', 'Nombre achats effectués','Catégorie de produit','seller_state']

Méthode du coude pour nombre de clusters :




Scatterplot obtenu (4 clusters) :



# Troisième étape : Algorithme de classification

Deuxièmes phases de test avec Kmeans simple et Transformation des variables :

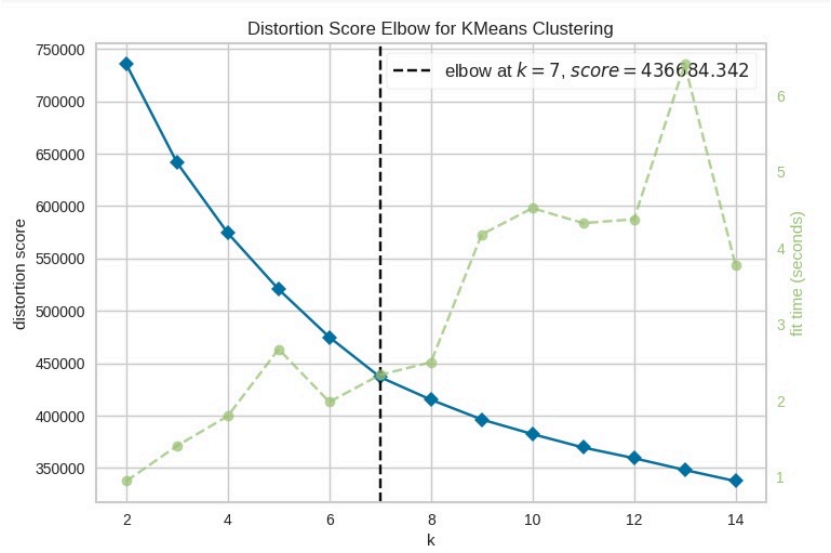
- Variables où l'on a appliqué la fonction logarithmique : ['price', 'Somme des achats', 'Nombre achats effectués']

- Variables utilisées au total : ['price','Somme des achats','Nombre achats effectués', 'Number_of_days_of_delay', 'days_diff','review_score','payment_installments','freight_value']

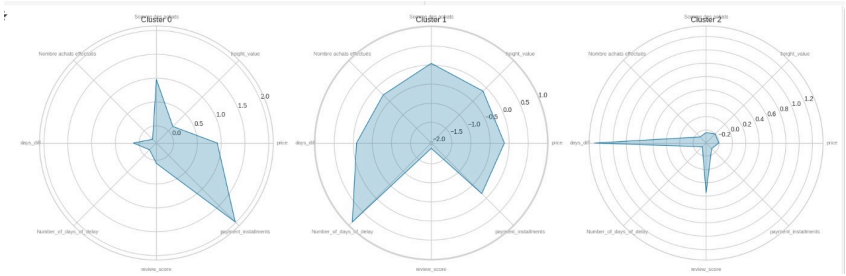
# Troisième étape : Algorithme de classification

Deuxièmes phases de test avec Kmeans simple et Transformation des variables :

Méthode du coude pour nombre de clusters :



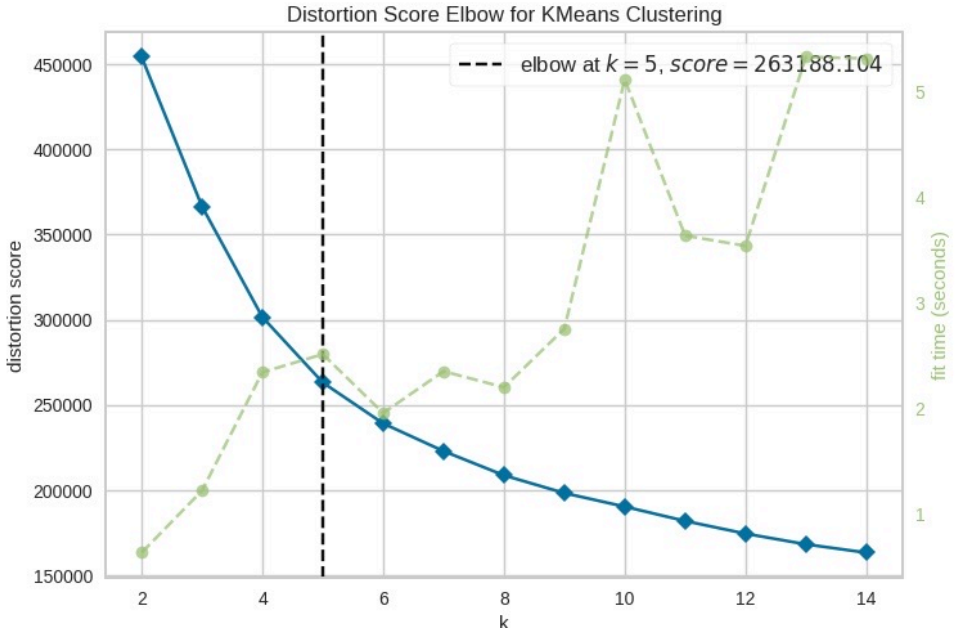

Exemples de Graphes sous forme de radars :

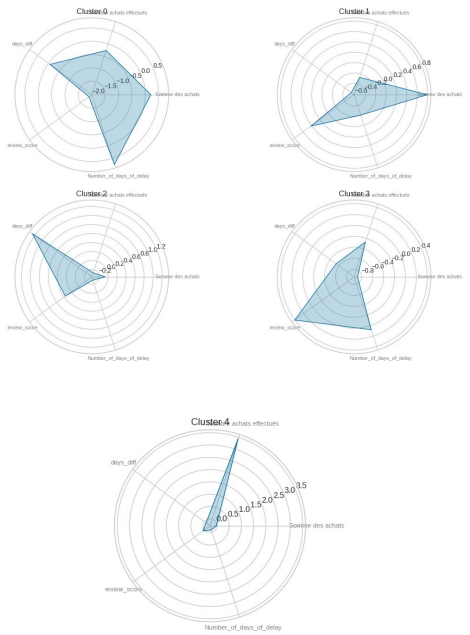



**Premier Test non concluant**

# Troisième étape : Algorithme de classification

Après plusieurs autres tests et suppression des variable 'payment_installments' et 'freight_value', le résultat optimal obtenu semble être de 5 clusters :



Néanmoins, la catégorisation des clusters n'est pas exploitable d'un point de vue client.

Passage à **6** clusters

# Troisième étape : Algorithme de classification

- Nombre de clusters choisi : **6**

- Catégorisation des groupes de clients :

1. Clients peu dépensiers, très satisfaits
2. Clients qui ont beaucoup acheté mais n'ont pas fait d'achats depuis longtemps
3. Clients qui achètent très souvent
4. Clients qui n'ont pas fait d'achats depuis longtemps et qui ont peu acheté
5. Clients qui ont beaucoup achetés et très satisfaits
6. Clients qui ont peu achetés et insatisfaits avec de nombreux jours de retards

# Troisième étape : Algorithme de classification

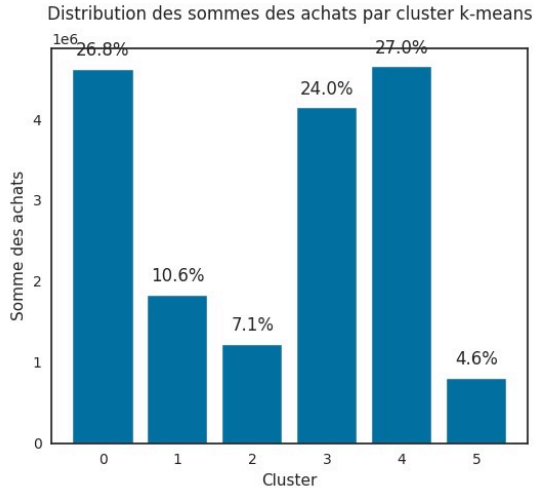
Résumé des résultats :

- Nombre de clusters choisi : **6**

- Catégorisation des groupes de clients :

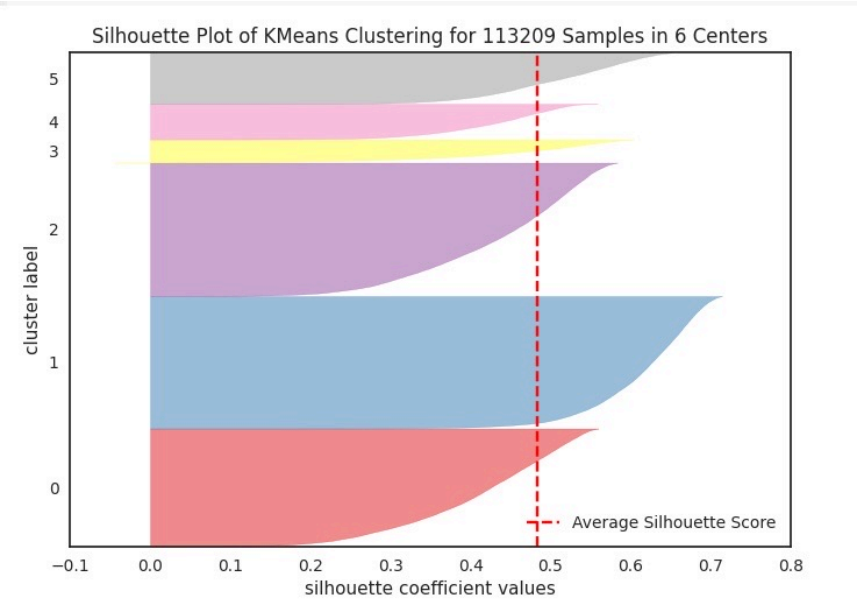
1. Clients peu dépensiers, très satisfaits
2. Clients qui ont beaucoup acheté mais n'ont pas fait d'achats depuis longtemps
3. Clients qui achètent très souvent
4. Clients qui n'ont pas achetés depuis longtemps
5. Clients qui ont beaucoup achetés et très satisfaits
6. Clients qui ont peu achetés et insatisfaits avec de nombreux jours de retards

CA par clusters :



# Troisième étape : Algorithme de classification

Coefficients de silhouette de l'algorithme choisi :



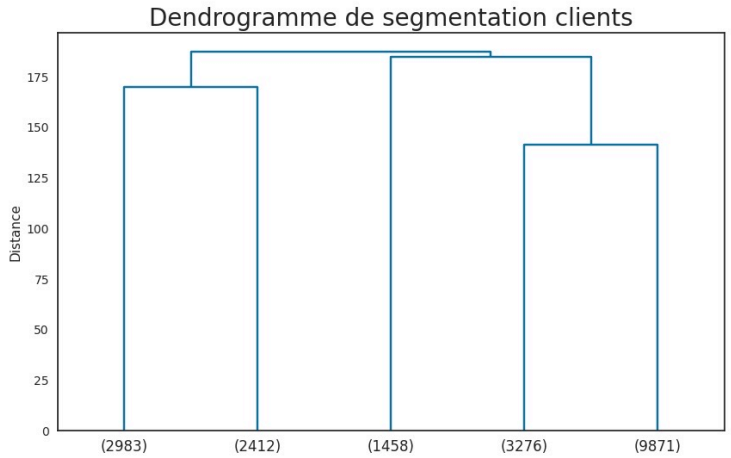

Observations :

La segmentation en **six clusters** montre une **cohésion modérée**, avec un score moyen de **silhouette de 0,48**. La majorité des clusters sont bien définis, mais certains pourraient être améliorés en termes de **séparation** et de **compacité**.

# Troisième étape : Algorithme de classification

Essais d'autres algorithmes :

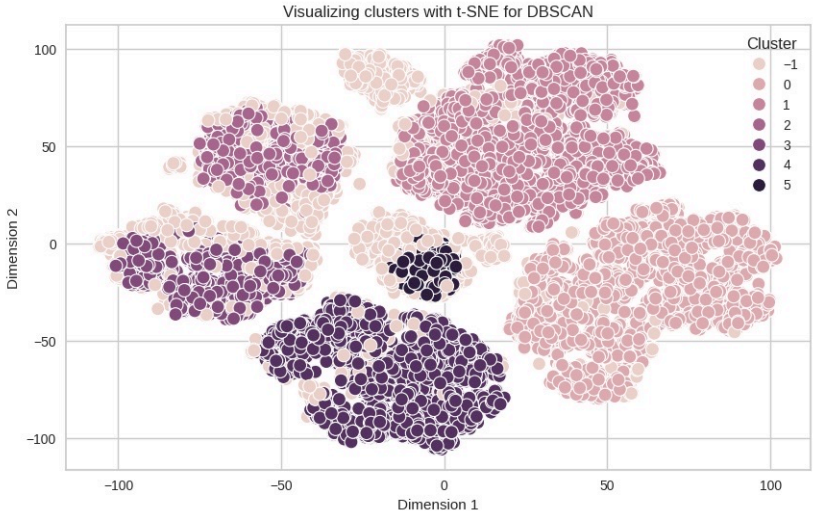
- Classification Ascendante Hiérarchique (CAH)




Observations :

- 1/5 du jeu de données car très chronophage
- Clusters relativement équilibrés exceptés pour le cinquième cluster (50% du jeu de données)

- DBSCAN




Observations :

- 6 clusters identifiés par l'algorithme
- Clusters relativement équilibrés
- Sur un échantillon de 20000, 3713 points identifiés comme du bruit (17% du jeu de données)

# **Quatrième étape :** Contrat de maintenance

**Rappel du contexte :**

- Olist souhaite une recommandation de fréquence à laquelle la segmentation doit être mise à jour pour rester pertinente, afin de pouvoir effectuer un devis de contrat de maintenance.

- Utilité : Adaptation du modèle aux nouvelles données et amélioration continue des performances

- Méthode utilisé : Ajusted Rand Index (ARI)

# **Quatrième étape :** Contrat de maintenance

## **Principe de l'ARI :**

L'**ARI** (Adjusted Rand Index) mesure la similarité entre deux partitions, servant à évaluer la qualité des clusters générés par un algorithme de clustering.

Comparaison de deux clusterings différents du même ensemble de données :

• Clustering de « référence", c'est-à-dire le clustering que l'on considère comme étant le plus précis ou le plus pertinent.

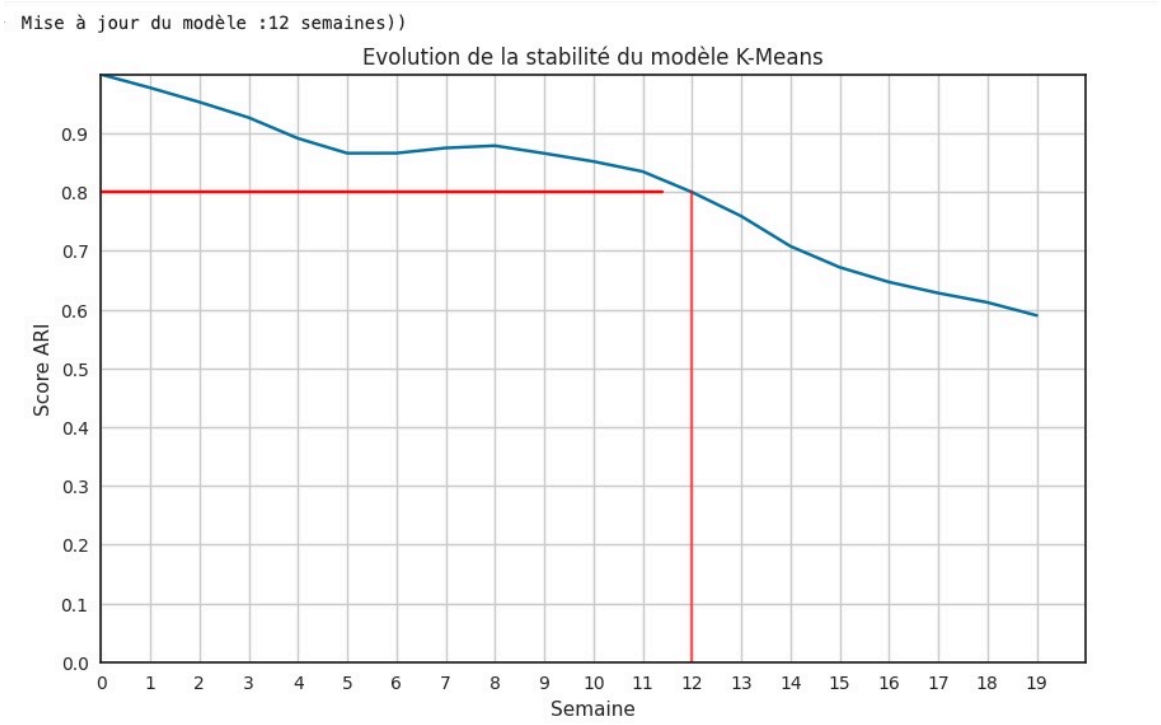
•Clustering de comparaison : ici, le jeu de données où les commandes ont débuté une semaine après la première commande réalisée au sein du dataset.

# **Quatrième étape :** Contrat de maintenance

L'ARI varie entre -1 et 1.

- ARI de 1 → Clusterings identiques.
- ARI de 0 → Clusterings aussi similaires que ce que l'on pourrait attendre du hasard.

ARI élevé → Forte concordance entre les clusters



# Conclusion

Segmentation en **6** clusters

Segments à privilégier :
- Clients peu dépensiers, très satisfaits (26,8% du CA)
- Clients qui n'ont pas achetés depuis longtemps (24% du CA)
- Clients qui ont beaucoup achetés et très satisfaits (27% du CA)

Contrat de maintenance à renouveler toutes les **12** semaines

Merci pour votre attention.