

## Projet 5 : Segmentez les clients d'un site de e-commerce

Lancelot LECLERCQ

21 janvier 2022

# Sommaire

1. Introduction
2. Analyse et transformation des données
3. Essais de différents modèles
4. Simulation de l'évolution de la classification
5. Conclusion

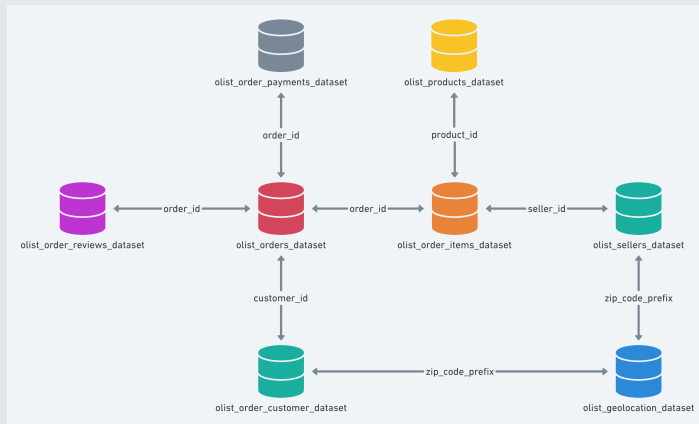
# Introduction

# Problématique

- Client : Olist, site de e-commerce
  - Souhaite effectuer une segmentation des clients
  - Comprendre les différents types d'utilisateurs
- Objectifs :
  - Fournir une description actionable de la segmentation et de sa logique pour une utilisation optimale
  - Faire une proposition de contrat de maintenance à partir de l'analyse de la stabilité de la classification au cours du temps

# Jeu de données

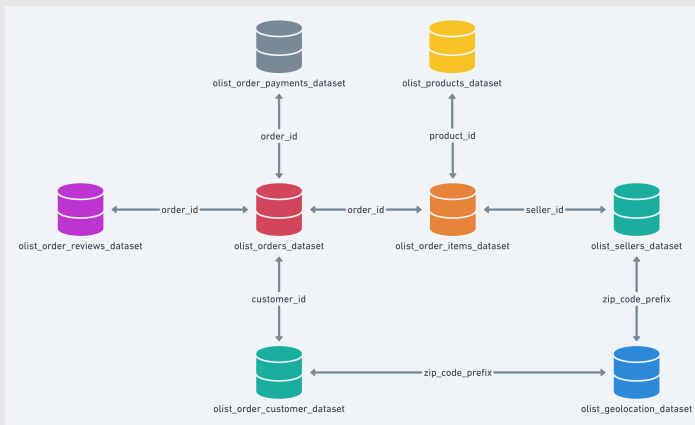
- Base de données anonymisée du site
- Données concernant :
  - les clients,
  - les vendeurs,
  - les commandes,
  - les produits vendus,
  - les commentaires et la satisfaction
- Données des années 2017 et 2018



# Analyse et transformation des données

# Analyse et transformation des données : structure de la base de données

- Fichier central contenant les commandes (olist\_orders\_dataset)
- Des identifiants permettent de rassembler les différents fichiers constituant la base de données
  - order\_id relie les données de paiement, les produits commandés et les données de notation aux commandes,
  - customer\_id relie les identifiants des clients aux commandes,
  - products\_id relie les données concernant les produits aux produits commandés,
  - seller\_id relie les données de vendeurs aux produits commandés,
  - zip\_code\_prefix relie les données de géolocalisation des acheteurs et des vendeurs
- Ces données vont nous permettre de calculer des variables plus intéressantes dans le cadre de cette segmentation des clients



# Analyse et transformation des données : calcul de nouvelles variables

- Études des clients = nécessité de regrouper les données de commandes par client
- Permet d'obtenir :
  - la date de la dernière commande
  - le prix moyen d'une commande d'un client
  - le nombre de commandes par client
  - le temps moyen entre deux commandes effectuées par un même client
  - la note moyenne données par un client
  - le nombre de produits par catégories de produits achetés par un clients
  - ⋮
- Utilisation des méthodes traditionnelles de marketing pour classifier les clients : classification RFM
  - Recency : la date à laquelle a été effectuée la dernière commande
  - Frequency : le nombre de commande
  - Monetary : le prix moyen d'une commandes d'un client



# Analyse et transformation des données : analyse des données

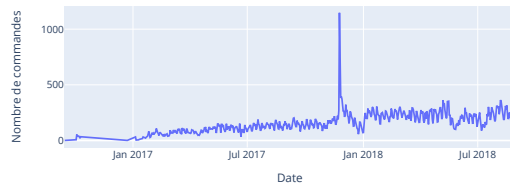
- Augmentation du nombre de commandes durant l'année 2017
- Développement après le lancement

- Pic d'achat au 24 novembre 2017 qui correspond au Black Friday

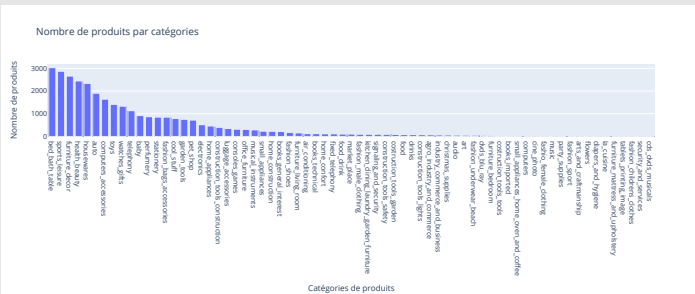
Nombre de commandes par mois



Nombre de commandes par jours



## Analyse et transformation des données : analyse des données



- Très grand nombre de catégories de produits



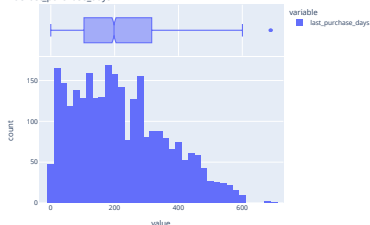
- Regroupement en 10 catégories

# Analyse et transformation des données : analyse des données

## Visualisation des données de RFM

### Recency

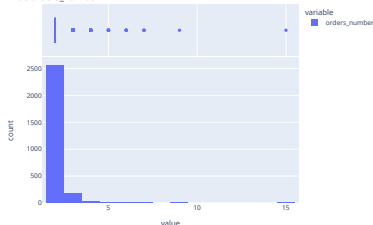
Histogramme et diagramme en boîte des données de last\_purchase\_days



- Beaucoup d'achats entre 0 et 200, mediane autour de 200

### Frequency

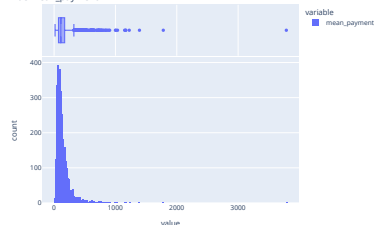
Histogramme et diagramme en boîte des données de orders\_number



- Majorité des clients ont effectués 2 achats mais quelques exeptions (9, 15)

### Monetary

Histogramme et diagramme en boîte des données de mean\_payment

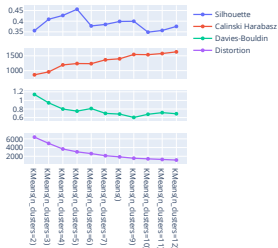


- Dépense médiane autour de 100 certains clients sont très dépensiers jusqu'à 3700

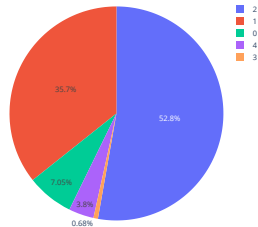
## Essais de différents modèles

# Essais de différents modèles : KMeans

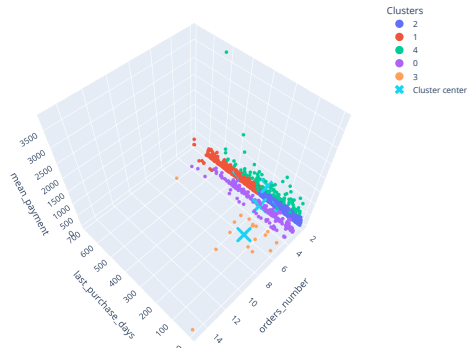
Visualisation des scores de classification selon le paramètre du modèle



Nombre de clients par clusters de l'algorithme KMeans



Clusters créés par KMeans

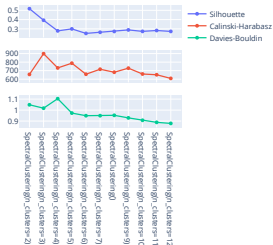


## - Les 5 clusters sont clairement interprétables

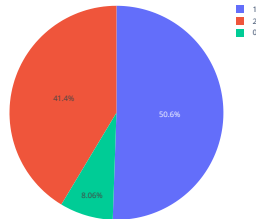
- 0 : clients qui ont fait 3 ou 4 achats d'une valeur moyenne < 450
- 1 : clients qui ont fait 2 achats il y a plus longtemps (environ plus de 250j) d'une valeur moyenne < 600
- 2 : clients qui ont fait 2 achats plutôt récemment (environ moins de 250j) d'une valeur moyenne < 350
- 3 : clients qui ont fait plus de 5 achats d'une valeur moyenne < 400
- 4 : client qui ont fait 2, 3 ou 4 achats de valeurs plus importante (environ > 350)

# Essais de différents modèles : SpectralClustering

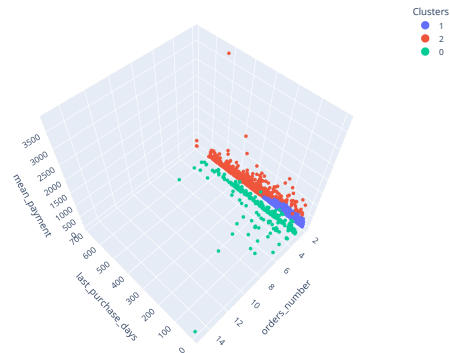
Visualisation des scores de classification selon le paramètre du modèle



Nombre de clients par clusters de l'algorithme SpectralClustering



Clusters créés par SpectralClustering

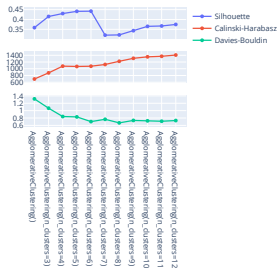


## - Les 3 clusters sont clairement interprétables

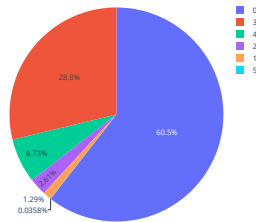
- 0 : clients qui ont fait plus de 3 achats d'une valeur moyenne < 700
- 1 : clients qui ont fait 2 achats plutôt récemment (environ moins de 250j) et d'une valeur moyenne < 400
- 2 : clients qui ont fait 2 achats il y a plus longtemps (environ plus de 250j) et/ou d'une valeur moyenne > 400

# Essais de différents modèles : AgglomerativeClustering

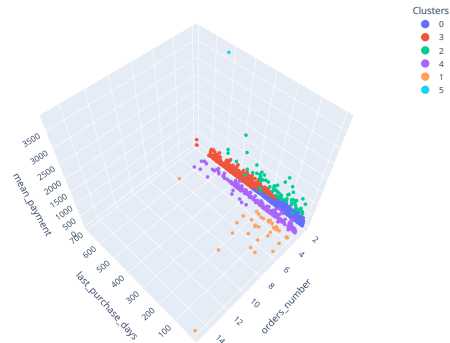
Visualisation des scores de classification selon le paramètre du modèle



Nombre de clients par clusters de l'algorithme AgglomerativeClustering



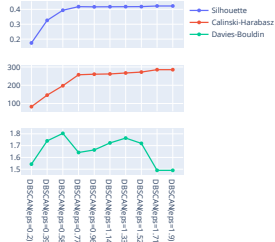
Clusters créés par AgglomerativeClusters



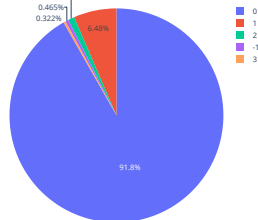
- Les 6 clusters sont plutôt bien définis mais se chevauchent légèrement plus que ceux de l'algorithme KMeans
  - 0 : clients qui ont fait 2 achats plutôt récemment (environ moins de 250j) d'une valeur moyenne < 500
  - 1 : clients qui ont fait plus de 4 achats d'une valeur moyenne < 450
  - 2 : clients qui ont fait entre 2 et 4 achats d'une valeur moyenne > 400
  - 3 : clients qui ont fait 2 commandes il y a plus longtemps (> 300j pour les montants les moins importants et > 150j pour les montants plus importants)
  - 4 : clients qui ont fait 3 achats ou 4 achats de valeurs plus importantes (environ > 400) ou il y a plus longtemps (> 250j)
  - clients qui ont fait 2 achats de valeurs très importantes (> 3500)

# Essais de différents modèles : DBSCAN

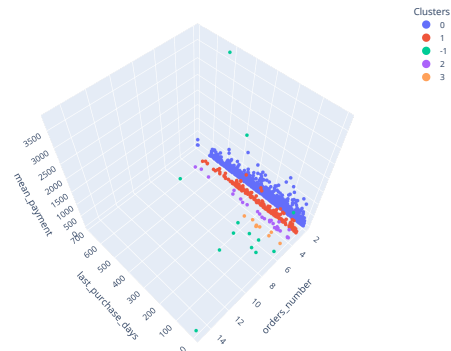
Visualisation des scores de classification selon le paramètre du modèle



Nombre de clients par clusters de l'algorithme DBSCAN



Clusters créés par DBSCAN



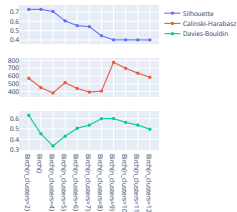
- Cet algorithme cherche lui même le nombre de cluster. Il en a créé 4 et éliminé quelques points. Les quatres clusters correspondent au nombre d'achats effectués

- 0 : clients qui ont fait 2 achats
- 1 : clients qui ont fait 3 achats
- 2 : clients qui ont fait 4 achats
- 3 : clients qui ont fait 5 achats

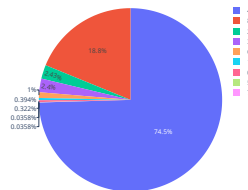


# Essais de différents modèles : Birch

Visualisation des scores de classification selon le paramètre du modèle

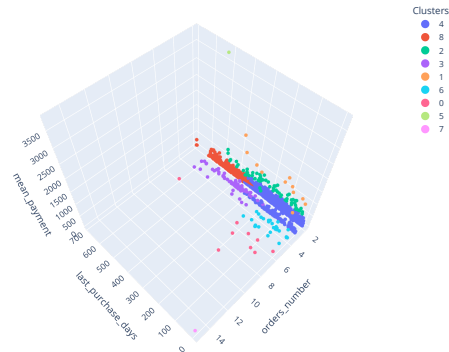


Nombre de clients par clusters de l'algorithme Birch



- Les 9 clusters sont plutôt bien définis mais cela commence à faire un grand nombre de clusters pour différencier des clients
  - 0 : clients ayant fait entre 6 et 9 commandes
  - 1 : clients ayant fait des commandes d'une valeur moyenne plutôt élevée (>800)
  - 2 : clients ayant fait 2 ou 3 commandes d'une valeur moyenne plutôt moyenne (entre 400 et 800)
  - 3 : clients ayant fait 3 ou 4 commandes d'une valeur moyenne plutôt basse (<400) et il y a plus longtemps (>300j)
  - 4 : clients ayant fait 2 ou 3 commandes d'une valeur moyenne plutôt basse (<500) et plus récemment (<300j)
  - 5 : clients ayant fait 2 commandes d'une valeur moyenne très élevée (>3500)
  - 6 : clients ayant fait 4 ou 5 commandes plutôt récemment (<250j)
  - 7 : clients ayant fait un très grand nombre de commandes (15)
  - 8 : clients ayant fait 2 commandes d'une valeur moyenne plutôt basse (<450) et il y a plus longtemps (>350j)

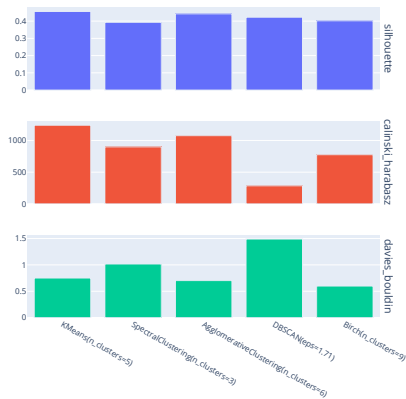
Clusters créés par Birch



# Essais de différents modèles : comparaison des modèles

- Plus le coefficient de silhouette est haut plus les clusters sont définis
- Il est calculé pour chaque objet et est composé de deux scores :
  - La distance entre cet objet et les autres objets contenu dans la même classe
  - La distance moyenne entre cet objet et ceux contenus dans la classe la plus proche
- Plus le score de Calinski-Harabasz est haut plus les clusters sont denses et bien séparés
- Calcul le ratio de la somme des dispersions inter-clusters et de celle des dispersions intra-clusters
- Plus le score de Davies-Bouldin est bas plus les clusters sont définis
- Calcul la similarité entre les clusters. Cette similarité compare la distance entre les clusters avec la taille des clusters

Comparaison des scores des modèles de classification



Au vu de ces données je choisis d'utiliser l'algorithme KMeans qui a à la fois un coefficient de silhouette et un score de Calinski-Harabasz élevés

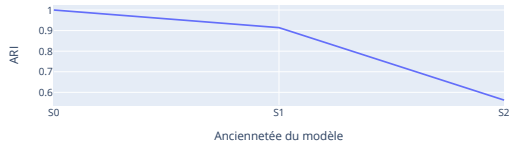
## Simulation de l'évolution de la classification

# Simulation de l'évolution de la classification : aspect technique

- Utilisation de l'algorithme KMeans à 5 clusters
- Utilisation du score ARI : Adjusted Rand Index
  - Compare la similarité entre les labels assignés à un même objet pour différentes méthodes de classification
- Sélection et ajout (incrémentation) des clients qui nous intéressent (plus de 2 commandes effectuées et reçues) par période donnée (mensuelle, trimestrielle, semestrielle)
- Comparaison pour chaque incrémentation avec les données les plus récentes pour les clients correspondants
  - L'observation de l'évolution de l'ARI va nous permettre de voir combien de temps les labels assignés restent pertinents

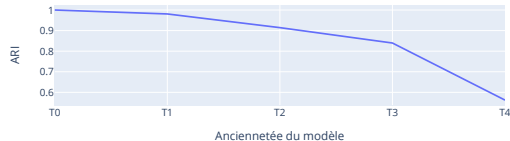
# Simulation de l'évolution de la classification : visualisation de l'évolution

ARI du modèle en fonction de l'ancienneté semestrielle du modèle

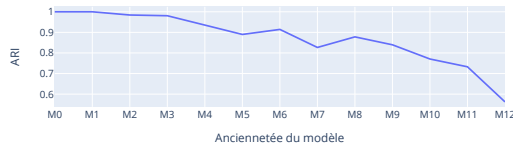


- ↑ Le premier semestre semble être encore assez bien corrélé ( $ARI > 0.9$ ) le second chute assez brutalement ( $ARI < 0.6$ )
- ↗ Le premier trimestre reste plutôt bien corrélé ( $ARI > 0.98$ ). La pente augmente ensuite jusqu'au 3<sup>e</sup> trimestre. On retrouve la chute importante au 4<sup>e</sup> trimestre
- Les trois premiers mois les données restent bien corrélées ( $ARI > 0.98$ ) mais après on observe une pente plus importante entre le 3<sup>e</sup> et le 11<sup>e</sup> mois avec une chute au 12<sup>e</sup> mois

ARI du modèle en fonction de l'ancienneté trimestrielle du modèle



ARI du modèle en fonction de l'ancienneté mensuelle du modèle



## Conclusion

- L'algorithme KMeans me paraît le plus pertinent car il est le plus performant pour un nombre de cluster raisonnable
- Il propose une classification ni trop complexe ni trop simple qui ne permettrait pas de bien comprendre le profil du client
- Idéalement renouvellement trimestriel afin de garder des données au plus près de la réalité (ARI > 0.98)
- Le renouvellement tous les 6 mois me semble rester pertinent car l'ARI reste autour de 0.9
- Par contre au-delà la chute du score s'accélère et au bout d'un an l'ARI passe en dessous de 0.6