

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

Lancelot LECLERCQ

24 janvier 2022

# Sommaire

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

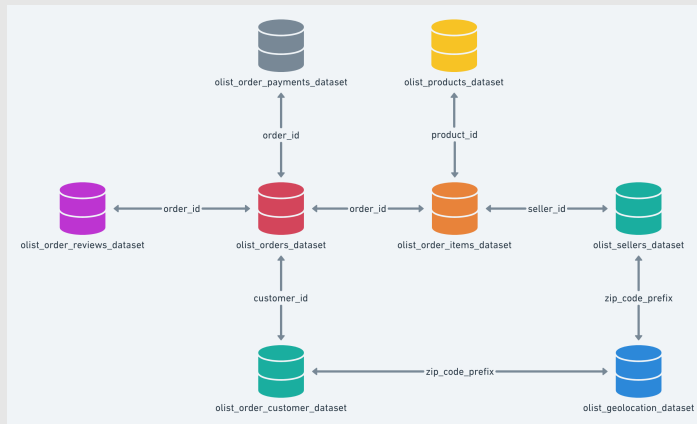
# Introduction

# 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 segmentation au cours du temps

# Jeu de données

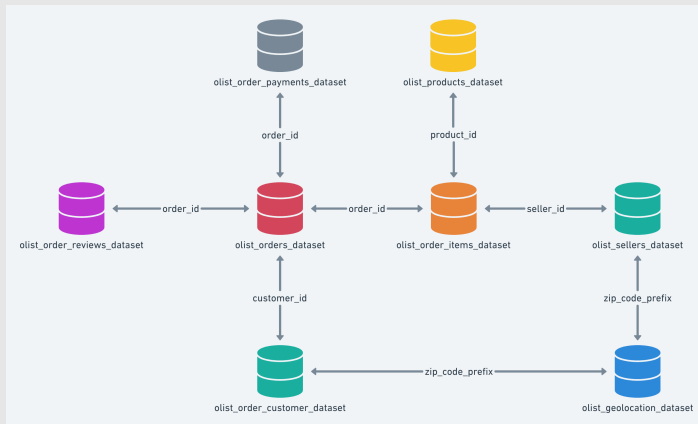
- Base de données anonymisée du site
- Données concernant :
  - les clients,
  - les vendeurs,
  - les commandes,
  - les produits vendus,
  - les commentaires et la satisfaction
- Données des années 2017 et 2018



# 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
- 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

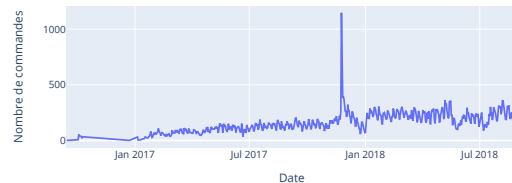
- Augmentation du nombre de commandes durant l'année 2017
- Développement après le lancement

- 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

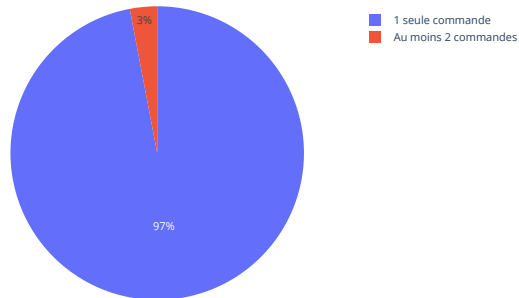
- Seul 3% des clients font au moins 2 commandes

TAB. : Nombre de clients par nombre de commandes réalisées

Nb commandes	Nb clients
1	90557
2	2573
3	181
4	28
5	9
6	5
7	3
9	1
15	1

- Le nombre de clients ayant réalisé plus de 3 commandes est très faible nous allons donc les supprimer

Proportion de clients en fonction du nombre de commandes effectuées

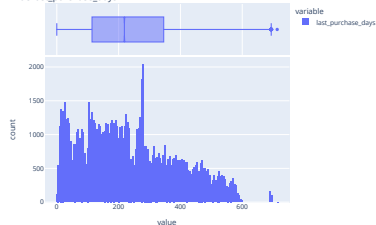


# Analyse et transformation des données : analyse des données

## Visualisation des données de RFM

### Recency

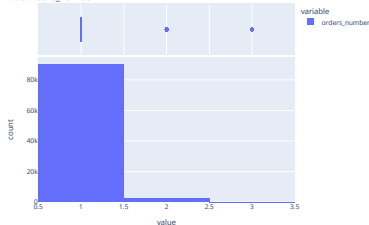
Histogramme et diagramme en boîte des données de last\_purchase\_days



- Beaucoup d'achats entre 0 et 250 jours, mediane autour de 250
- S'explique peut être du fait que le lancement du site commence seulement 2 ans avant la fin de nos données

### Frequency

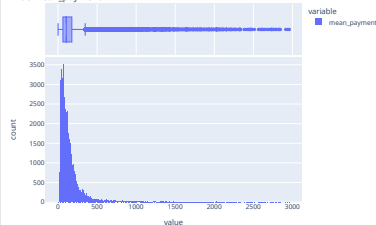
Histogramme et diagramme en boîte des données de orders\_number



- Majorité des clients ont effectués 1 seul achat mais quelques uns font 2 ou 3 commandes

### Monetary

Histogramme et diagramme en boîte des données de mean\_payment



- Dépense médiane autour de 100 certains clients sont très dépensiers jusqu'à 3000

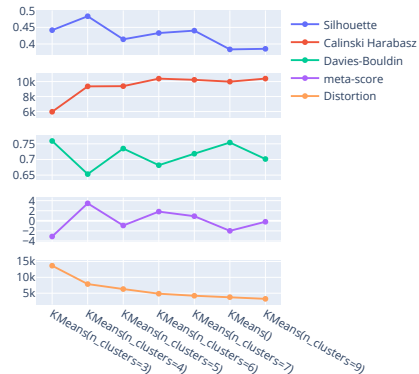
## Essais de différents modèles

# Essais de différents modèles : KMeans

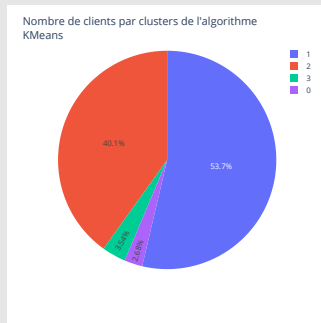
## Description des méthodes d'évaluations des classifications

- 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

Visualisation des scores de classification selon le paramètre du modèle

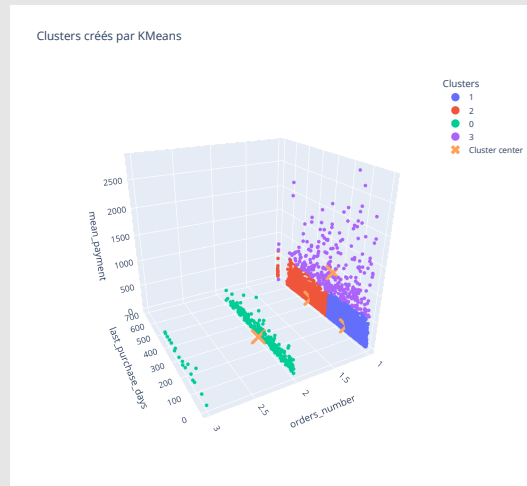


# Essais de différents modèles : KMeans



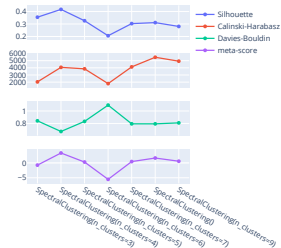
## - Les 4 clusters sont clairement interprétables

- 0 : clients qui ont fait au moins 2 commandes
- 1 : clients qui ont fait 1 commande d'une valeur moyenne < 500 et plutôt récemment (environ moins de 300j)
- 2 : clients qui ont fait 1 commande il y a plus longtemps (> 300j) et d'une valeur plus faible (< 5000)
- 3 : clients qui ont fait 1 commande de plutôt grande valeur (> 500)

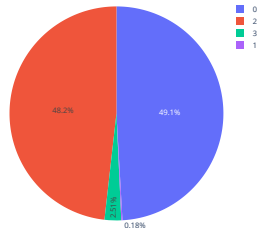


# Essais de différents modèles : SpectralClustering

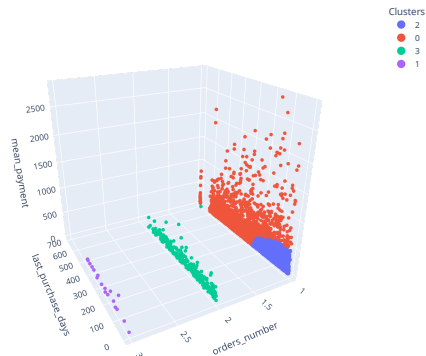
Visualisation des scores de classification selon le paramètre du modèle



Nombre de clients par clusters de l'algorithme SpectralClustering



Clusters créés par SpectralClustering

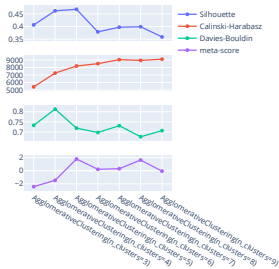


## - Les 4 clusters sont clairement interprétables

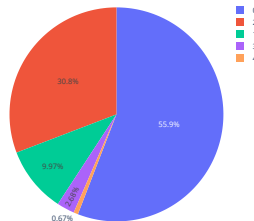
- 0 : clients qui ont fait 1 commande il y a plus longtemps ou plutôt récemment d'une d'une valeur > 500
- 1 : clients qui ont fait 3 commandes
- 2 : clients qui ont fait 1 commande plus récemment (< 250j) et d'une valeur moins importante (< 500)
- 3 : clients qui ont fait 2 commandes

# Essais de différents modèles : AgglomerativeClustering

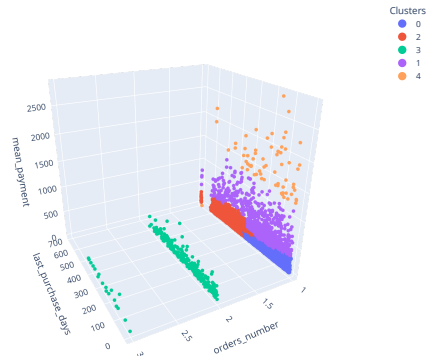
Visualisation des scores de classification selon le paramètre du modèle



Nombre de clients par clusters de l'algorithme AgglomerativeClustering



Clusters créés par AgglomerativeClusters



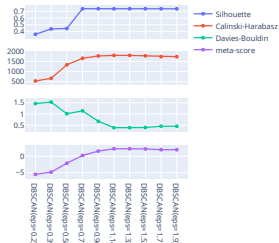
## - Les 5 clusters sont plutôt bien définis

- 0 : clients qui ont fait 1 commande plutôt récemment (< 300j) d'une valeur moyenne < 300
- 1 : clients qui ont fait 1 commande d'une valeur moyenne > 200 mais < 1400
- 2 : clients qui ont fait 1 commande il y a plus longtemps (> 300j)
- 3 : clients qui ont fait 2 ou 3 commandes
- 4 : clients qui ont fait 1 commande d'une valeur importante (> 1400)

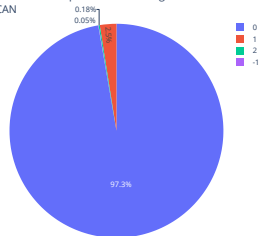


# Essais de différents modèles : DBSCAN

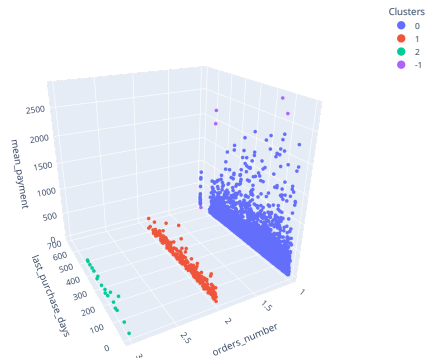
Visualisation des scores de classification selon le paramètre du modèle



Nombre de clients par clusters de l'algorithme DBSCAN



Clusters créés par DBSCAN

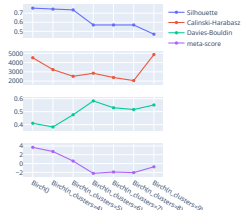


- Cet algorithme cherche lui même le nombre de cluster. Il en a créé 3 et éliminé quelques points. Les trois clusters correspondent au nombre d'achats effectués

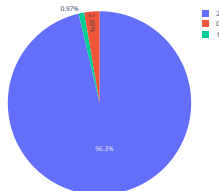
- 0 : clients qui ont fait 1 achat
- 1 : clients qui ont fait 2 achats
- 2 : clients qui ont fait 3 achats

# Essais de différents modèles : Birch

Visualisation des scores de classification selon le paramètre du modèle



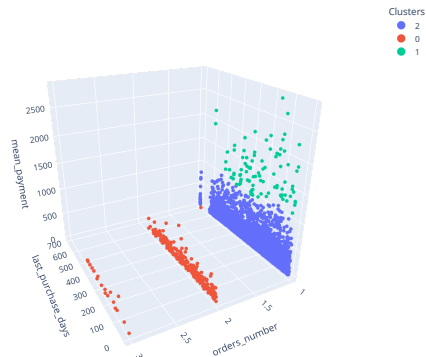
Nombre de clients par clusters de l'algorithme Birch



## - Nous avons 3 clusters

- 0 : clients ayant fait au moins 2 commandes
- 1 : clients ayant fait 1 commande d'une valeur > 1000
- 2 : clients ayant fait 1 commande d'une valeur < 1000

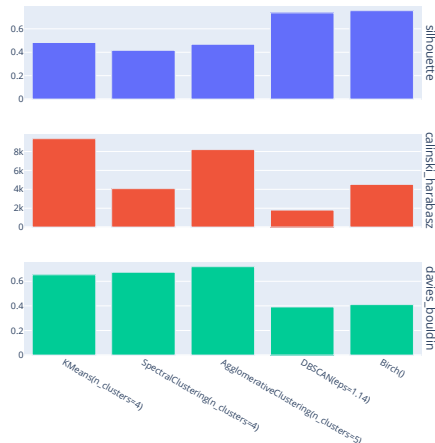
Clusters créés par Birch



# Essais de différents modèles : comparaison des modèles

- KMeans est l'algorithme ayant le plus haut score de Calinski-Harabasz
- KMeans possède un score de silhouette inférieur à d'autre algorithme
- KMeans possède une meilleur répartition des clients au sein des groupes
  - Évite des petits clusters
- Au vu de ces données je choisis d'utiliser l'algorithme KMeans qui a à la fois un coefficient de silhouette moyen et un score de Calinski-Harabasz élevés mais aussi une bonne répartition du nombre d'objets par classe.

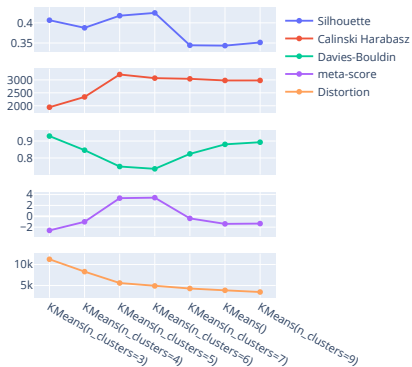
Comparaison des scores des modèles de classification



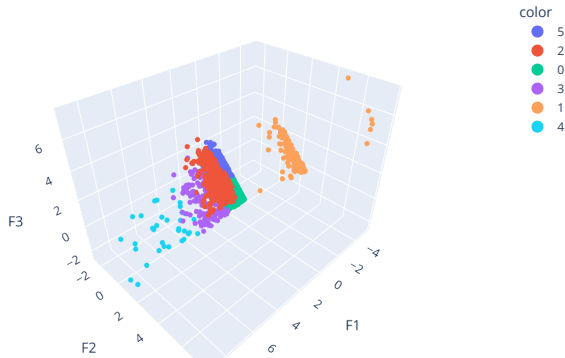
## Modélisation avec ajouts de la satisfaction

# Modélisation avec ajouts de la satisfaction : visualisation des clusters en PCA

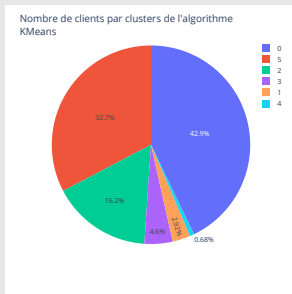
Visualisation des scores de classification selon le paramètre du modèle



Visualisation 3D de la PCA



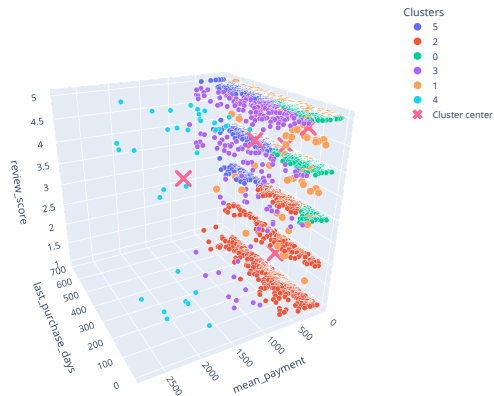
# Modélisation avec ajouts de la satisfaction : visualisation des données brutes



## - Les 6 clusters sont

- 0 : clients qui ont fait 1 commande récente (< 300j) d'une valeur moyenne < 400 et plutôt satisfaits (note entre 3 et 5)
- 1 : clients qui ont fait 2 ou 3 commandes
- 2 : clients qui ont fait 1 commande d'une valeur < 600 et peu satisfaits (note entre 1 et 3)
- 3 : clients qui ont fait 1 commande il y a plus longtemps (> 300j) et d'une valeur plus faible (< 250)
- 4 : clients qui ont fait 1 commande de grande valeur (> 1200)
- 5 : clients qui ont fait 1 commande il y a plus longtemps (> 300j) d'une valeur moyenne < 400 et plutôt satisfaits (note entre 3 et 5)

Clusters créés par KMeans



## Simulation de l'évolution de la classification

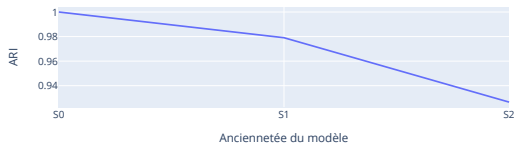
# Simulation de l'évolution de la classification : aspect technique

- 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 (incrémentations) 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



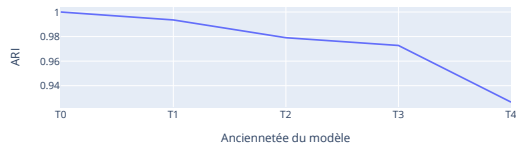
# Simulation de l'évolution de la classification : visualisation de l'évolution

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



- ↑ On observe une baisse du score tout au long de l'année avec une légère accélération (pente plus importante) sur la deuxième moitié
- ↗ Le premier trimestre reste plutôt bien corrélé. La pente augmente légèrement ensuite jusqu'au 3<sup>e</sup> trimestre. On retrouve la chute plus importante au 4<sup>e</sup> trimestre
- Les 6 premiers mois les données restent bien corrélées ( $ARI > 0.98$ ) mais après on observe des variations dont des chutes du score pour certains mois (M7) et après 10 mois la chute est brutale

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



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



## Conclusion

- Choix de l'algorithme :
  - L'algorithme KMeans me paraît le plus intéressant car il est le plus performant pour un nombre de cluster raisonnable
  - Il propose une classification pertinente qui permet de bien comprendre le profil du client
- Fréquence de mise à jour du modèle :
  - Je recommande un renouvellement trimestriel afin de garder des données au plus près de la réalité
  - Le renouvellement tous les 6 mois me semble rester pertinent