

# Segmentation des clients d'un site de e-commerce **olist**

E. Campos / Aout 2023



# Contexte et problématique

## Segmentation de clients e-commerce

Client

olist

- **Activité** : propose à des entrepreneurs des solutions de ventes sur les marketplaces en ligne pour élargir leur base clients
- **Origine** : Brésil
- **Année de création** : 2015
- **Valorisation** : \$1.5 Md\* (après la levée de fonds de \$186m en décembre 2021)

Mission

- Proposer une **segmentation des clients** de Olist qui pourra être utilisée au quotidien par le service marketing pour ses **campagnes de communication**.
- Cette segmentation doit permettre de comprendre les différents types d'utilisateurs, via leur **comportement** et leurs **données personnelles**.

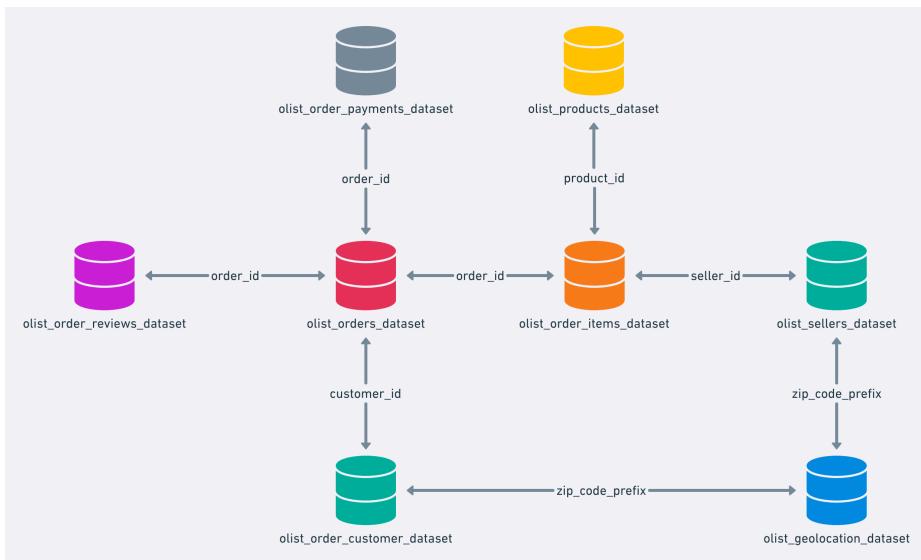
Livrable

- ❖ **Description actionnable de la segmentation** et de sa logique sous-jacente.
- ❖ **Proposition d'un contrat de maintenance** basée sur une analyse de la stabilité des segments au cours du temps.

\* Source: [techcrunch.com](https://techcrunch.com)

# Présentation et nettoyage des données

- Les données utiles à l'analyse sont issues de 8 bases de données différentes.
- Ces bases de données sont joignables via la **clé primaire relative au numéro de commande** (« order\_id »)



## Retraitements avant fusion des datasets

- Suppression de clients avec des coordonnées géographiques incohérentes (6)
- Simplification des catégories d'articles (passage de 72 à 16)
- Imputation de valeurs manquantes :
  - 279 coordonnées géographiques clients :
    - 229 : moyenne des coordonnées des clients situés dans la même ville
    - 50 : moyenne des coordonnées des clients situés dans le même Etat
  - 14 coordonnées géographiques vendeurs : moyenne des coordonnées des vendeurs de la même ville
  - 8 dates de livraison : date de livraison prévue
  - 2 dates de remise au transporteur : date d'approbation de la commande + médiane de la durée entre l'approbation et la remise au transporteur
  - 14 dates d'approbation de la commande : date de la commande + médiane de la durée entre la commande et son approbation
  - 13 traductions manquantes : recherche dans dictionnaire
  - 610 nombre de photo / description d'articles : remplacés par 0 en supposant qu'une valeur manquante signifie qu'il n'y a pas de photos ou de description

## Retraitements après fusion des datasets

- Suppression des commandes qui ont un statut différent que « delivered »
- Imputation des valeurs manquantes :
  - Les caractéristiques de paiement d'une commande (commande qui n'était pas dans le dataset payment) : imputation par le mode
  - Il reste alors 5122 valeurs manquantes (0,08% du jeu de données), relatives à des caractéristiques produits (nom de la catégorie, taille,...) et aux notes/commentaires laissées par les clients (cas où le client n'a pas répondu à l'enquête de satisfaction)

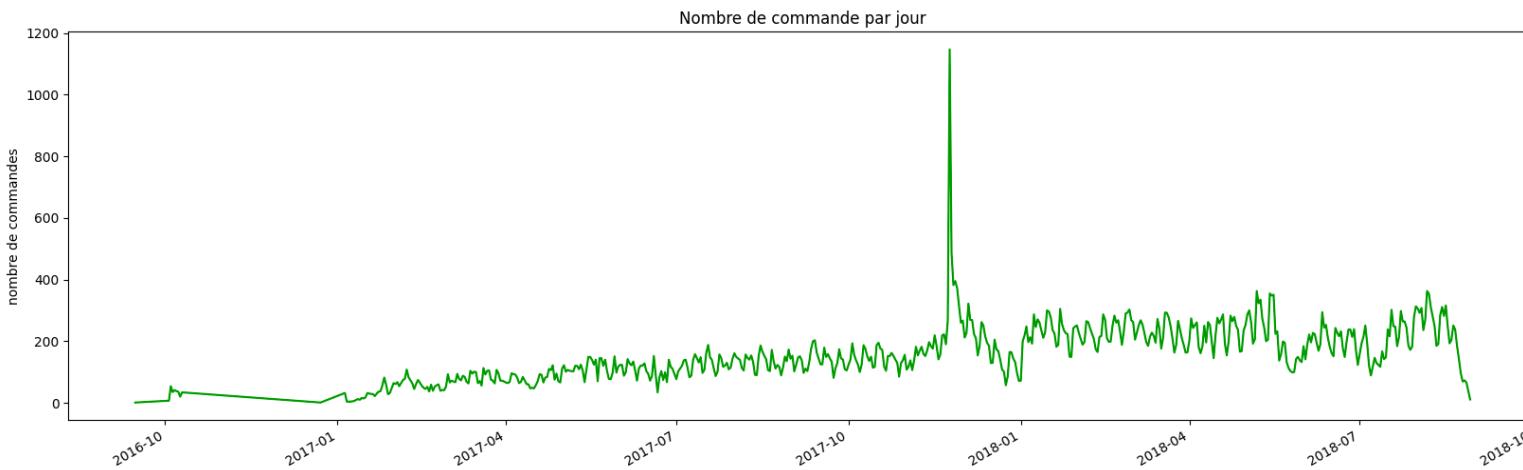
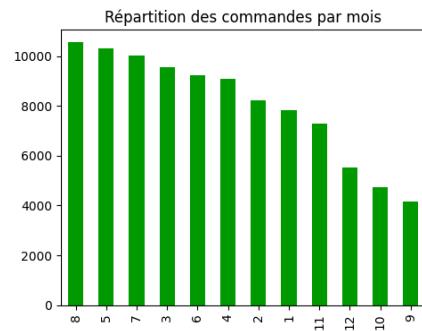
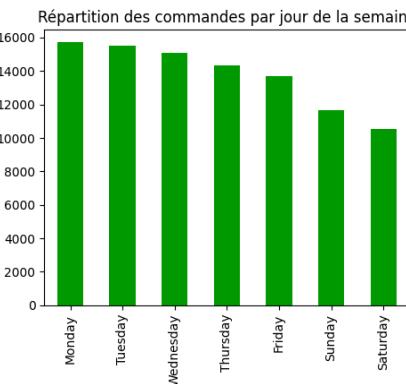
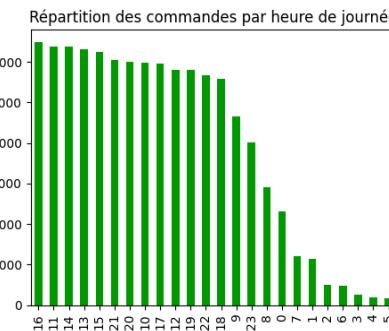
# Création de variables

Variables pour imputation
Variables pour EDA
Variables pour segmentation

Nom de la variable	Description
<code>order_to_delivery_d</code>	Durée (en jours) entre la commande et la livraison
<code>order_to_approval_h</code>	Durée (en heures) entre la commande et son approbation
<code>approval_to_carrier_d</code>	Durée (en jours) entre l'approbation de la commande et sa remise au transporteur
<code>carrier_to_customer_d</code>	Durée (en jours) entre la remise de la commande au transporteur et la livraison au client
<code>real_vs_estimated_d</code>	Durée (en jours) entre la date de livraison effective et la date prévue
<code>delay</code>	Variable booléenne qui indique si la livraison a eu du retard par rapport à la date prévue
<code>nb_of_items</code>	Nombre d'articles par commande
<code>freight_as_pc_price</code>	Prix de la livraison exprimé en pourcentage du prix de l'article
<code>high_freight</code>	Variable booléenne qui indique si le prix de la livraison dépasse le prix de l'article
<code>nb_of_transactions</code>	Nombre de transactions par commande (nombre de séquences de paiement différentes)
<code>volume_cm3</code>	Volume d'un produit (produit de la longueur, largeur et hauteur)
<code>limit_respect</code>	Variable booléenne qui indique si la limite de la date d'envoie (shipping) a été respectée
<code>delivery_to_survey_d</code>	Durée entre la réception d'une commande et l'envoie d'une enquête de satisfaction
<code>delivery_to_review_d</code>	Durée entre la réception d'une commande et la rédaction d'un avis par le client
<code>survey_to_review_d</code>	Durée entre l'envoi d'une enquête de satisfaction et la rédaction de l'avis par le client
<code>review_comment_title &amp; review_comment_message</code>	Variables booléennes qui indiquent si la revue laissée par le client a un titre ou un message (plutôt que la longueur de ce titre ou message)
<code>distance_sell_cust_km</code>	Distance en km entre le vendeur et le client
<code>order_total_value</code>	Valeur totale de chaque commande (et non uniquement par séquence de paiement)
<code>nb_of_items_bought</code>	Nombre d'article acheté par client
<code>telephony, pets, sport...</code>	Nombre d'article acheté par catégorie par client (16 variables différentes) => méthode préférée au OneHotEncoding
<code>nb_comments</code>	Nombre de commentaires rédigés par client
<code>nb_delay</code>	Nombre de retard subi par client
<code>favorite_purchase_hour, day, month</code>	Heure, jour et mois d'achat le plus fréquent par client

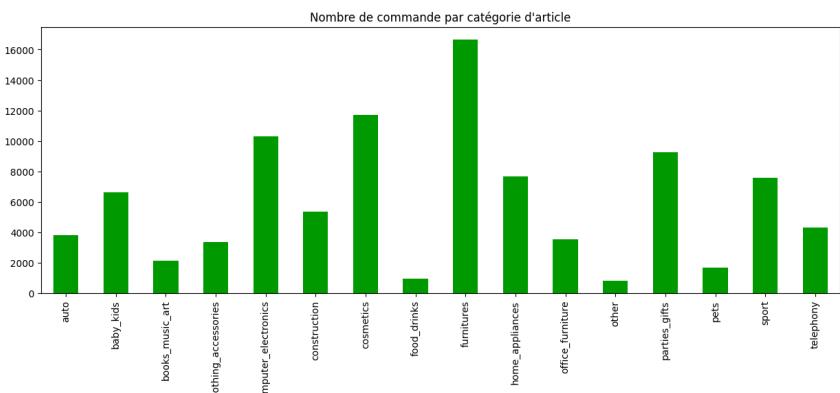
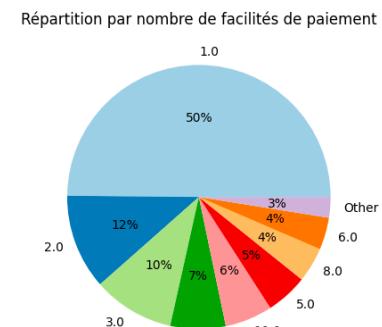
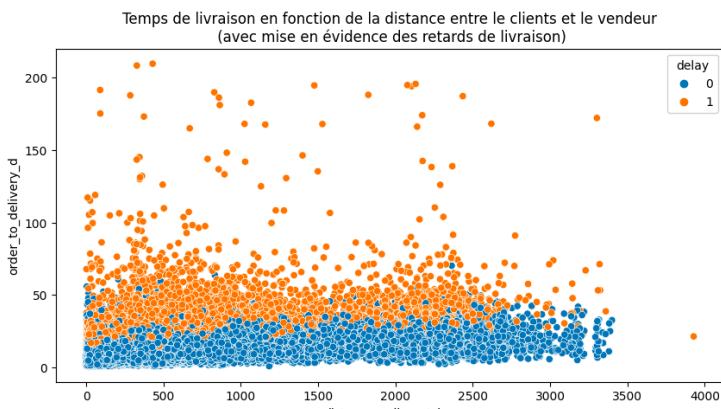
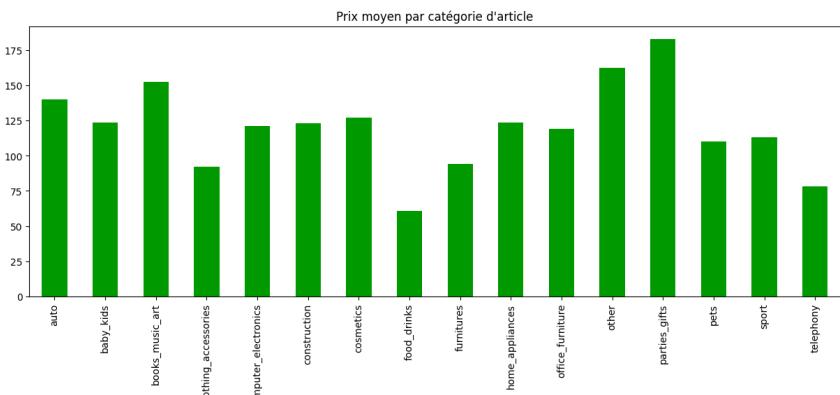
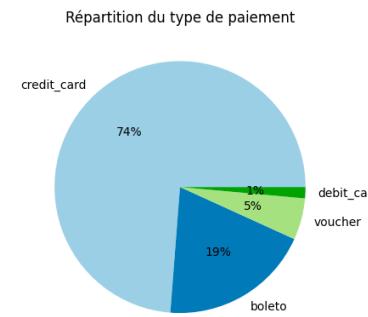
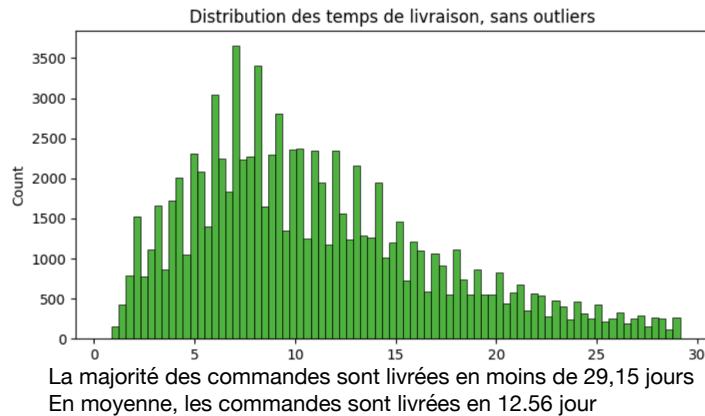
# Exploratory Data Analysis (1/3)

- Première commande : 15 septembre 2016
- Dernière commande : 29 aout 2018
- Période couverte : 713 jours
- Nombre de commande unique : 96 748
- Nombre de client unique : 93 358
- Habitude de consommation:
  - Mois d'août, mai et juillet,
  - Début de semaine (du lundi au mercredi),
  - Entre 10h et 22h
- La valeur des commandes ne varient pas significativement selon le moment où la commande a été effectuée

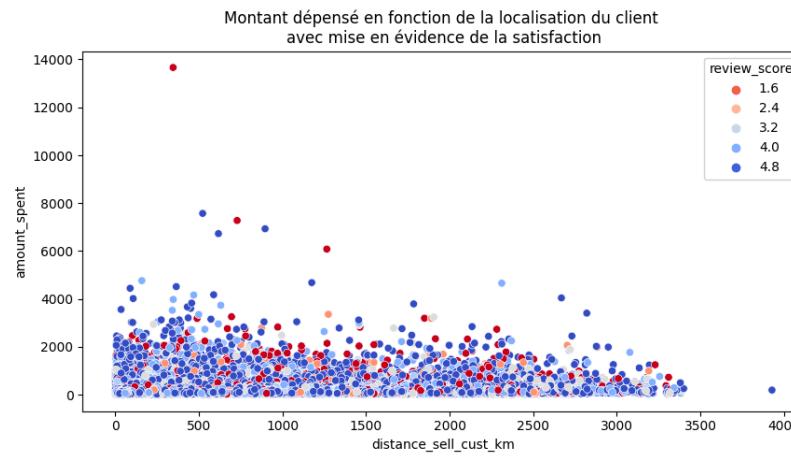
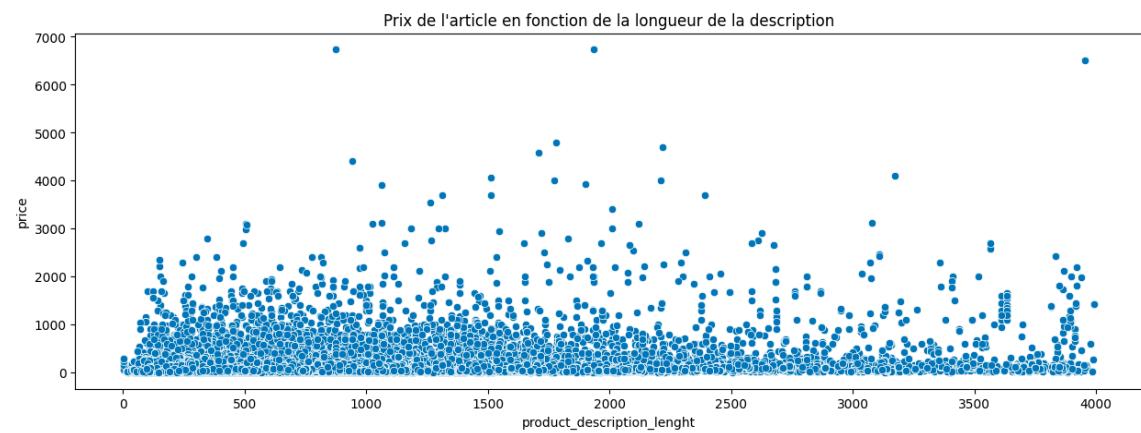
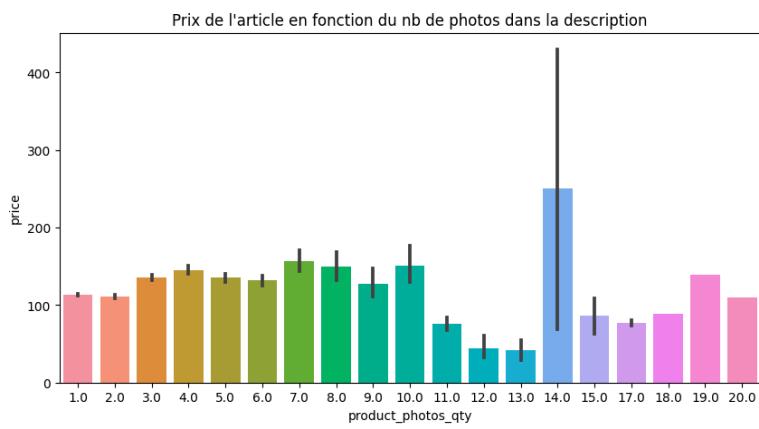
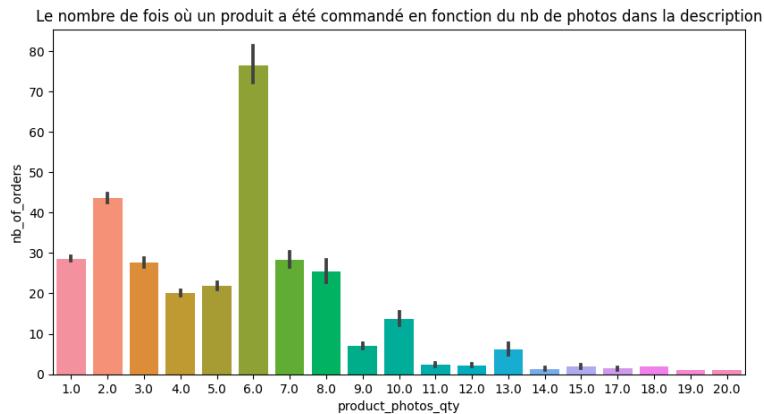


- Moyenne de 135,31 commandes par jour
- Pic de commande le 24 novembre 2017 : 1147 commandes (1,19% des commandes du dataset)
- Pas de commandes enregistrées entre octobre 2016 et janvier 2017

# Exploratory Data Analysis (2/3)



# Exploratory Data Analysis (3/3)



order_to_delivery_d	carrier_to_customer_d	0.922340
delivery_to_review_d	survey_to_review_d	0.900337
payment_value	order_total_value	0.877478
purchase_month	delivery_month	0.838478
payment_sequential	nb_of_transactions	0.834503
volume_cm3	product_weight_kg	0.806085
order_item_id	nb_of_items	0.793771
price	payment_value	0.737532
customer_lat	distance_sell_cust_km	0.706139
price	order_total_value	0.680800
freight_as_pc_price	high_freight	0.670454
carrier_to_customer_d	delivery_to_survey_d	0.628120
order_to_delivery_d	delivery_to_survey_d	0.625068
Freight_value	product_weight_kg	0.610803
real_vs_estimated_d	delay	0.598066

# Méthode 1

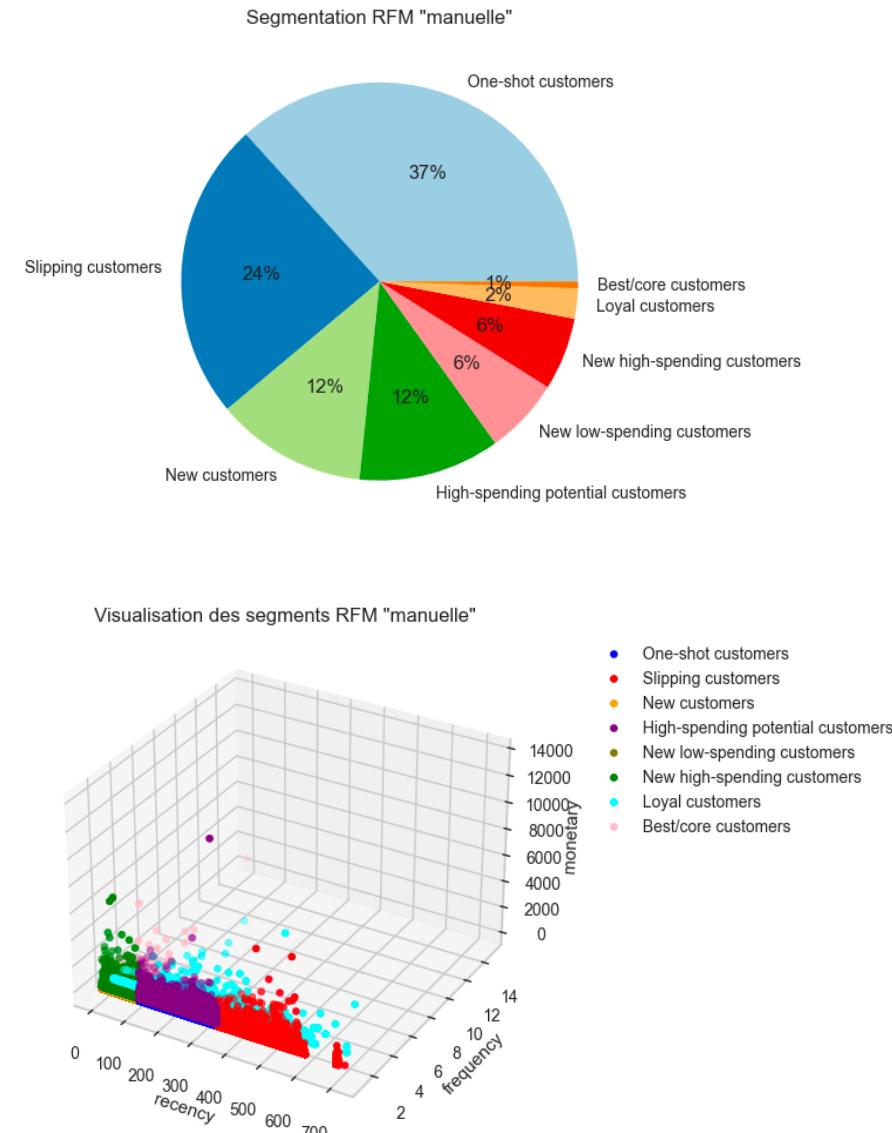
## RFM « manuelle »

- La méthode de segmentation **RFM** est très utilisées dans le domaine de la vente directe : elle permet de tenir compte du comportement d'achat des clients et de leur potentiel.
- Cette segmentation repose sur trois critères :
  - Récence** : nombre de jour depuis la dernière commande
  - Fréquence** : nombre d'achat effectué sur une période donnée
  - Montant** : somme totale dépensée par un client sur une période donnée

Méthodologie :

- Calcul de R\*, F et M
- Attribution d'une note entre 1 et 4 pour chaque variable sur la base de l'appartenance à un quartile (4 étant la meilleure note)
- Calcul d'un score global comme la concaténation des 3 scores R, F et M (ex: une note de 1 dans chaque dimension abouti à un score de « 111 »)
- Distinction des clients sur la base de leur score
  - ★ 444 : best/core customer
  - ★ 414 : new high-spending customer
  - ★ ...

\* R a été calculée à partir de la date de la dernière commande du dataset, soit le 29 aout 2018



# Méthode 2 - modèles (1/3)

## RFM avec des méthodes non supervisées

**Apprentissage non supervisé** (données non labellisées)

- Les modèles détectent des regroupements « seuls »

**Instance based learning**

- Les modèles généralisent en s'appuyant sur des mesures de similarité

**Préparation des variables**

- Les trois variables ayant des unités de mesure différentes ont des échelles de valeur différentes.
- Les algorithmes de clustering reposent sur des mesures de distance** : il est donc nécessaire de **scaler les features** pour améliorer la performance des modèles
- Différentes méthodes de mise à l'échelle ont été testées : la méthode « MinMax » appliquée aux 3 variables génère les meilleurs résultats

**Modèles testés**

### ★ KMean

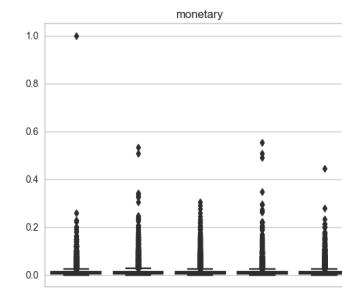
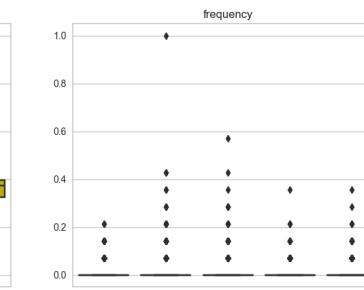
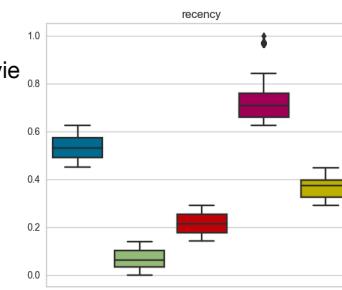
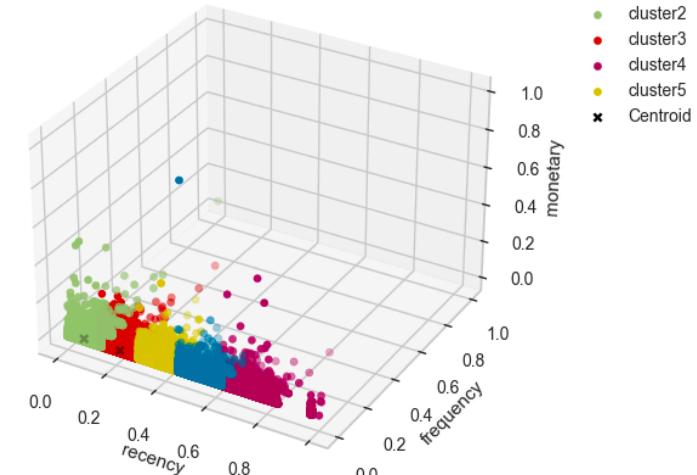
- Modèle qui cherche à diviser les datapoints en k groupes homogènes et compacts.
- Il fonctionne de manière itérative à partir de l'initialisation aléatoire des k centroïdes
- L'algorithme converge toujours\*, mais pas forcément sur la meilleure solution (optimum local) : tout dépend de l'initialisation des centroïdes
- Optimisation des hyperparamètres :
  - Via l'analyse combinée de l'inertie et des scores silhouette, Calinski Harabasz et Davies Bouldin
    - Nombre de clusters optimal k=5 (cf. slide suivant)
  - Via une « Grid Search » :
    - Méthode d'initialisation des clusters : « k-means ++» ou « random »
    - Nombre maximum d'itération de l'algorithme pour une exécution : 200, 300, 400

\* la distance quadratique moyenne entre un point et son centroïde ne peut que diminuer à chaque itération

cluster	
2	26.23
4	23.85
1	21.28
0	16.32
3	12.32

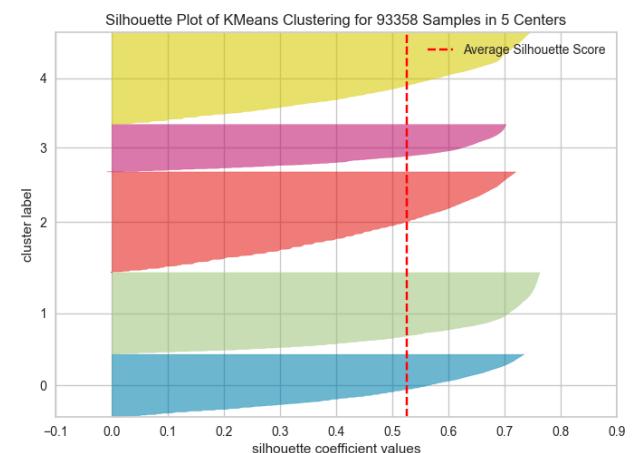
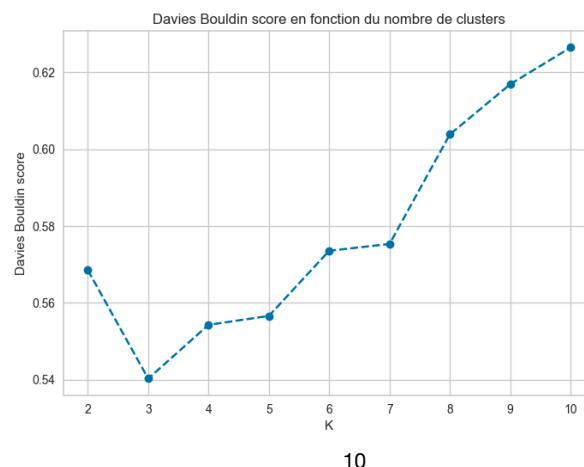
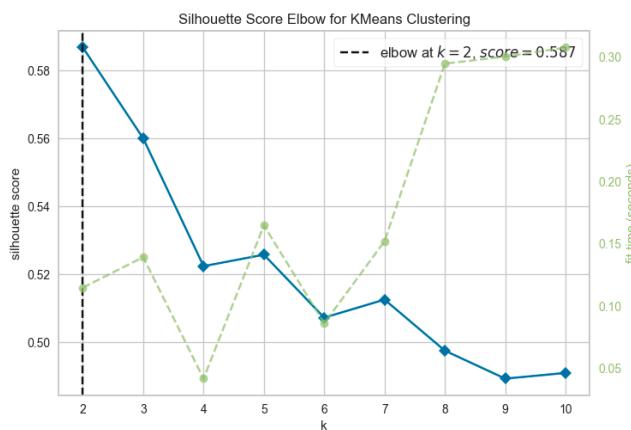
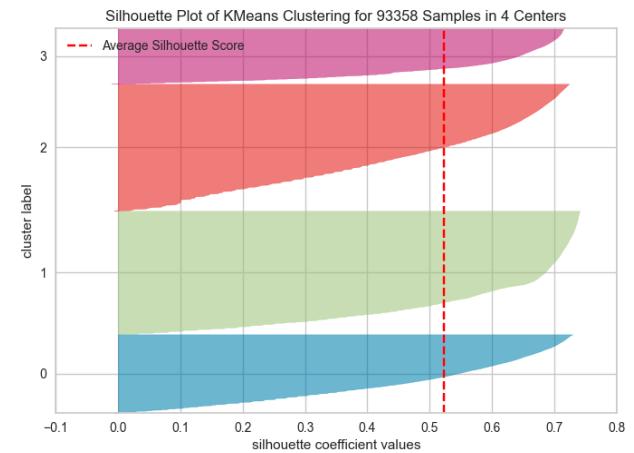
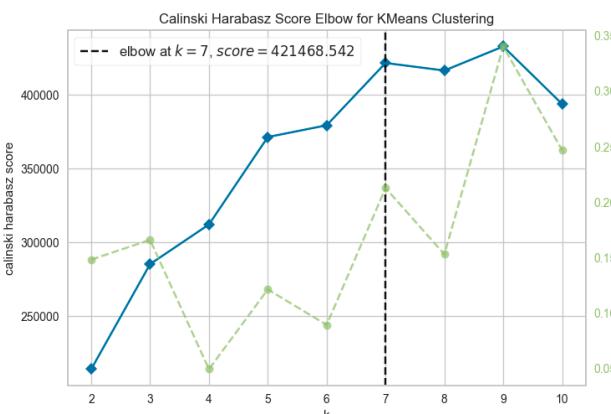
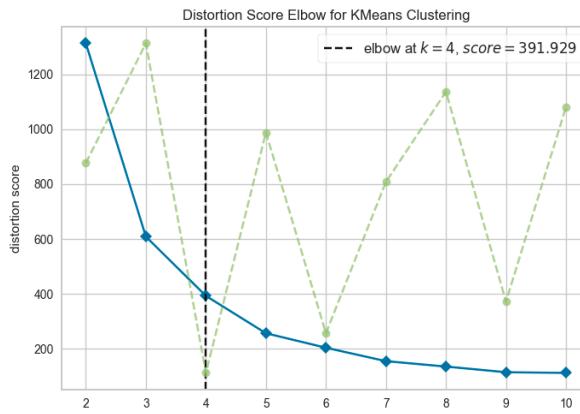
--- KMeans ---  
 pour 5 clusters  
 Inertia: 255.60  
 Silhouette score: 0.53  
 Calinski Harabasz score: 371,288.40  
 Davies Bouldin score: 0.56

RFM segmentation



# Détail des scores obtenus

## Nombre de clusters optimal - Kmeans



# Méthode 2 - modèles (2/3)

## RFM avec des méthodes non supervisées

### Modèles testés (suite)

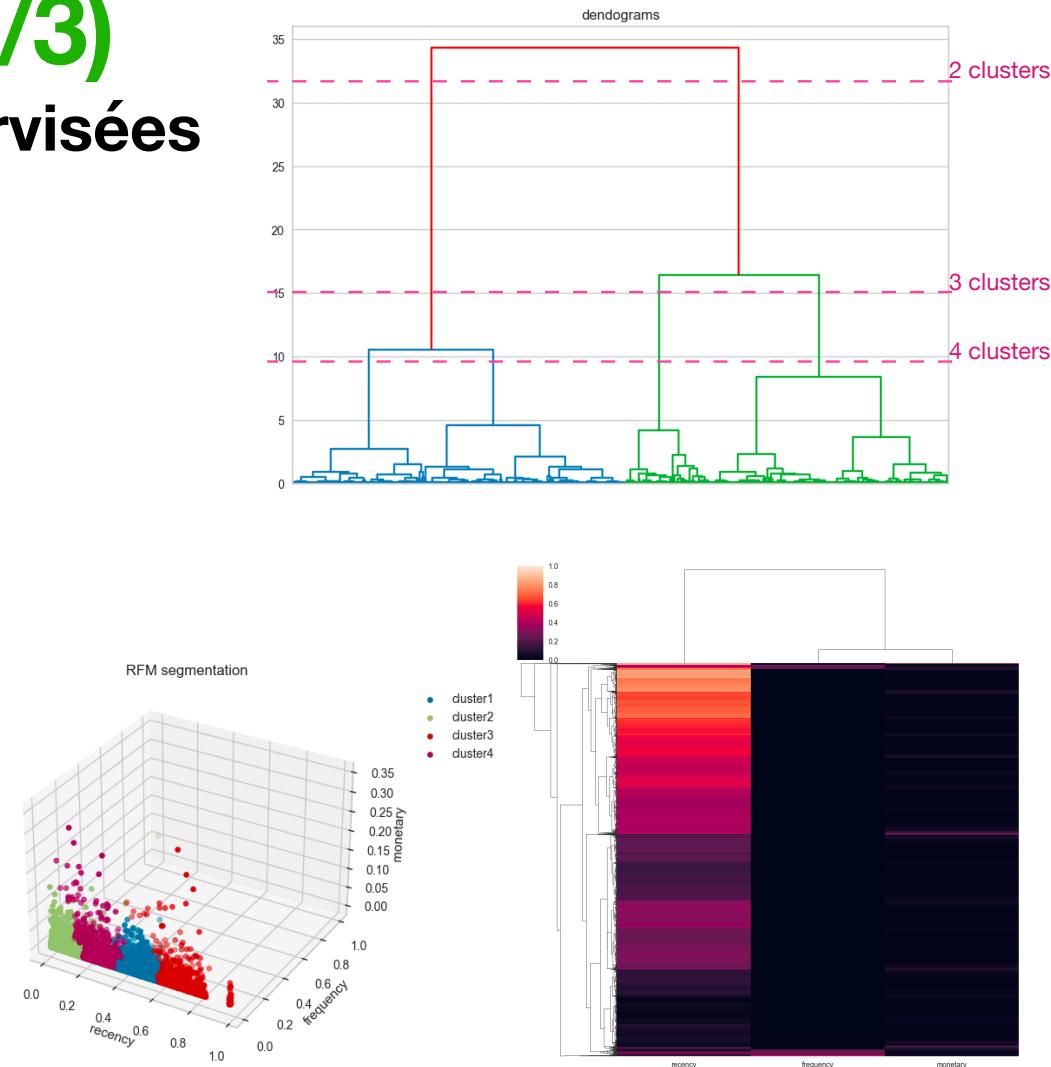
#### ★ Hierarchical clustering

- L'algorithme commence par considérer chaque datapoint comme étant son propre cluster, puis fusionne les points les plus proches\* de manière successive, jusqu'à aboutir au nombre de clusters souhaité
- Le nombre de cluster optimal\*\* ( $k=4$ ) a été sélectionné via
  - l'analyse combinée de l'inertie et des scores silhouette, Calinski Harabasz et Davies Bouldin
  - l'analyse visuelle du dendrogramme
- Inconvénient : l'algorithme a besoin de  $O(n^2)$  de mémoire et  $O(n^3)$  de runtime. Il ne fonctionne pas bien avec un nombre important de données. Notre dataset est composé de + de 93k données.
- Nous avons testé l'algorithme sur un subset de données. Nous obtenons des résultats assez proches de ceux du KMeans, mais (1) nous n'avons utilisé que 20% des données dispo et (2) le résultat n'est pas stable car variable en fonction de données du subset.

--- Hierarchical clustering ---	
pour 4 clusters	cluster
	0 37.77
Silhouette score: 0.51	2 30.70
Calinski Harabasz score: 60,412.06	1 21.13
Davies Bouldin score: 0.56	3 10.40

\* en l'occurrence nous avons appliqué la mesure de distance « ward » par défaut qui consiste à fusionner les clusters de telle façon que la variance intra-cluster augmente le moins (cela génère souvent des clusters de tailles homogènes)

\*\* le choix des autres hyperparamètres du modèle n'a pas été optimisé dans la mesure où ce dernier est inadapté pour traiter notre cas d'espèce



# Méthode 2 - modèles (3/3)

## RFM avec des méthodes non supervisées

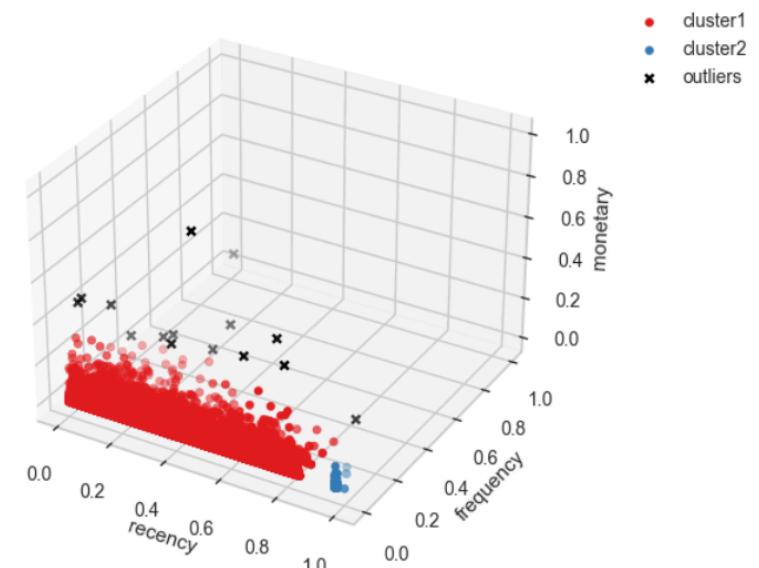
### Modèles testés (suite)

#### ★ DBSCAN (*density-based spatial clustering application with noise*)

- Algorithme qui définit des clusters sur la base de la densité des régions
    - Pour chaque datapoint, l'algorithme compte le nombre d'occurrences localisées à une distance ***epsilon***
    - Si le voisinage compte au moins ***min\_samples*** d'occurrences, le datapoint d'origine est considéré comme étant situé dans une région dense
    - Toutes les occurrences situées dans le voisinage de ce datapoint sont alors considérées comme appartenant au même cluster
    - Une occurrence qui n'est pas située dans une région dense est une anomalie (outlier)
  - Avantage : pas besoin de préciser un nombre de cluster au départ
  - Inconvénient : algorithme qui fonctionne bien si les clusters sont séparés par des régions à faible densité
- ➡ Ce n'est pas le cas du jeu de données OLIST
- Modèle non adapté et résultats peu pertinents
  - Hyperparamètres\* des résultat présenté :  $\text{eps}=0,1$  et  $\text{min\_samples}=10$

```
--- DBSCAN ---
estimated nb of clusters: 2
estimated nb of noise points: 15
Silhouette coef: 0.49
Calinski Harabasz score: 1170.38
Davies Bouldin score: 1.28
```

RFM segmentation

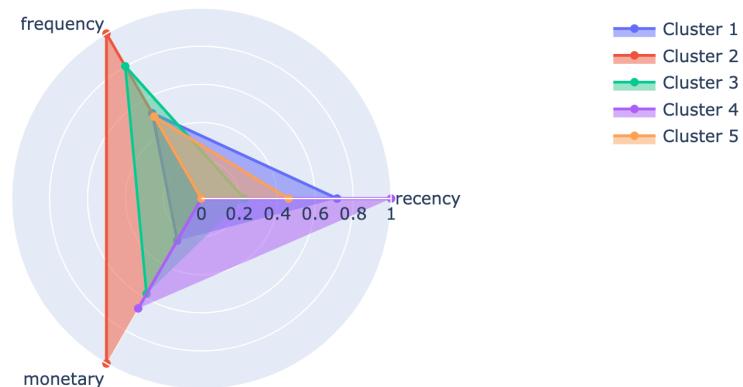


\* le choix des hyperparamètres du modèle n'a pas été optimisé dans la mesure où ce dernier est inadaptée pour traiter notre cas d'espèce

# Méthode 2 - résultats & interprétation Kmeans

## RFM avec des méthodes non supervisées

Comparaison des moyennes des variables par clusters



- **Cluster 1 (16%)**: clients dont la dernière commande date d'un an, pour des montants peu élevés, et qui achètent relativement peu fréquemment. Ce sont des **clients opportunistes** qui achètent pour profiter de bonnes affaires.  
➡ **Intensifier la communication pendant les périodes de solde et/ou prévoir des opérations spéciales avec des offres adaptées**
- **Cluster 2 (21%)**: clients dont la dernière commande date de moins de 2 mois, pour des montants élevés et qui achètent relativement fréquemment. Ces sont des **client à choyer**.  
➡ **Prévoir des mécanismes de fidélisation et de promotion ciblée (ventes privées, avant premières...) pour soutenir l'engouement de ces clients**
- **Cluster 3 (26%)**: clients dont la dernière commande date de 5 mois, pour des montants moyennement élevés et qui commandent de manière relativement récurrente. Ce sont des **clients à fidéliser** : ils connaissent déjà le service et n'hésitent pas à l'utiliser pour dépenser des sommes non négligeables.  
➡ **Comprendre pourquoi le client ne consomme par plus régulièrement au moyen de questionnaires et proposer des promotions ciblées pour relancer la relation**
- **Cluster 4 (13%)**: clients dont la dernière commande date de plus de 16 mois, pour des montants plutôt élevés mais qui commandent de manière très peu fréquente. Ce sont des **clients à relancer** : leur potentiel de dépense est élevé mais ils n'ont pas commandé depuis un certain moment  
➡ **Comprendre pourquoi le client n'a plus commandé depuis plus d'un ans. Est-ce du à une mauvaise expérience (qualité de service, état de l'article, délai de livraison...)?**
- **Cluster 5 (24%)**: clients dont la dernière commande date de 8 mois, pour des montants très peu élevés et qui commandent de manière peu fréquente. Ces **clients ont été perdus**.  
➡ **Concentrer les efforts marketing sur les autres segments identifiés**

# Méthode 3

## RFM « augmentée »

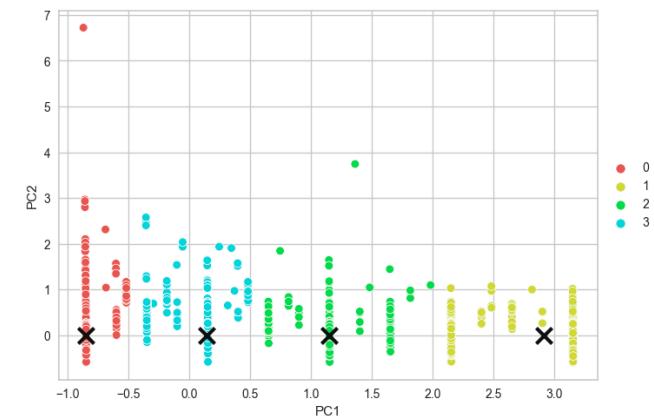
- En plus des variables R, F et M, nous avons ajouté d'autres variables susceptibles de nous renseigner sur les comportements d'achat et la satisfaction des clients:
  - Nombre d'articles achetés
  - Nombre d'article acheté par catégorie
  - Volume et poids moyen des articles achetés
  - Note moyenne
  - Nombre de commentaires rédigés
  - Durée entre le passage de la commande et la livraison
  - Distance entre le client et le vendeur
  - Nombre de retard subis
  - Heure, jour et mois d'achat préféré
  - Nombre de versement
  - ...
- Nous avons testé un ensemble de combinaisons de variables, en prenant notamment en compte (1) les corrélations entre les variables et (2) le caractère discriminant des variables.
- Les meilleurs résultats ont été obtenus avec les variables : **R, F, M, la note moyenne octroyée, le volume moyen des articles achetées**
- Cette combinaison de variables aboutit au meilleur compromis : score, nombre de cluster, interprétabilité, stabilité à l'initialisation

```
--- KMeans ---
pour 4 clusters
Inertia: 11,085.05
Silhouette score: 0.69
Calinski Harabasz score: 405,493.09
Davies Bouldin score: 0.47
```

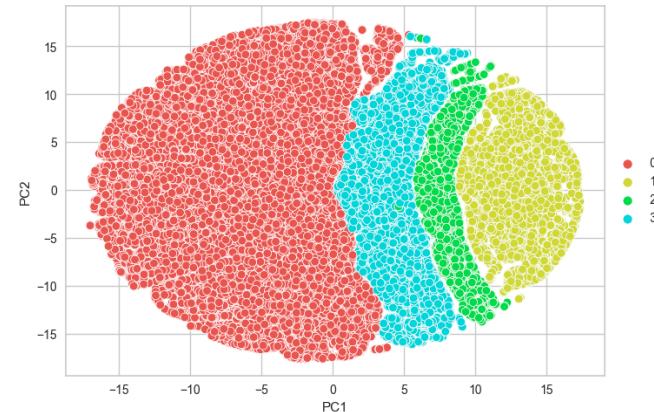
cluster	
0	58.77
3	19.98
1	12.64
2	8.60

Evaluation de la stabilité des clusters à l'initialisation  
 Index Rand ajusté moyen pour 25 itération : **0,95**

Visualisation  
PCA



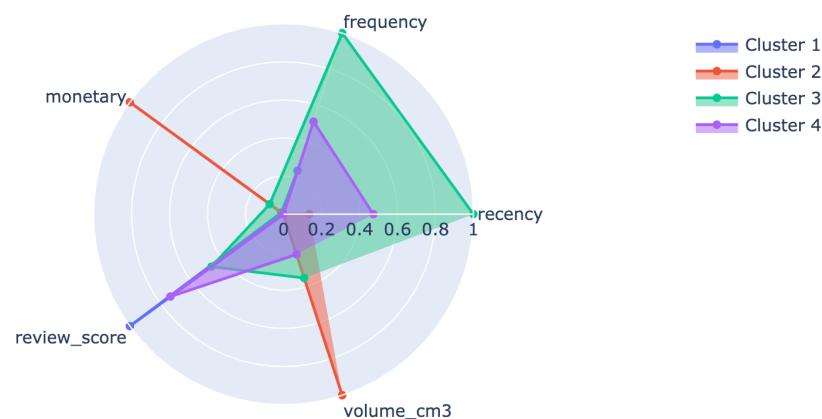
Visualisation  
TSNE  
(perplexity=5)



# Méthode 3 - résultats & interprétation

## RFM « augmentée »

Comparaison des moyennes des variables par clusters



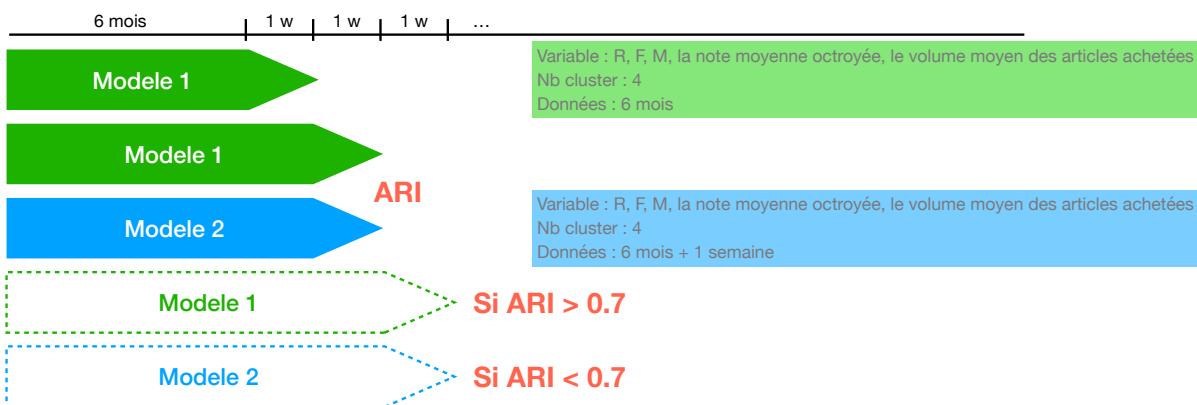
- **Cluster 1 (59%)** : clients qui dépensent peu, mais l'ont fait très récemment et sont satisfaits de leur expérience. Ils achètent plutôt des petits objets. On remarque également qu'ils vivent relativement proches des vendeurs et bénéficient ainsi de temps de livraison courts (10 jours). **Clients à convaincre**: le client a essayé le service et est satisfait, il s'agit maintenant de créer une habitude de consommation  
➡ Profiter du momentum du dernier achat et communiquer sur les avantages d'Olist (large gamme de produit, facilité de paiement...).
- **Cluster 2 (12%)** : clients qui dépensent beaucoup et achètent de gros objets lourds. Leur dernière commande est relativement récente. Par contre, ils ne sont pas du tout satisfaits. On remarque que ces clients souffrent de temps de livraison très élevés (plus de 20 jours), et qu'ils habitent en moyenne assez loin des vendeurs (=675km). Ces clients ont également laissé des commentaires qui risquent d'être négatifs (étant donné leur niveau de satisfaction) et cela peut créer des effets de propagation. **Clients dont il faut réparer la relation pour sauvegarder le potentiel de dépense et enrayer le développement de sentiments négatifs**.  
➡ Envoyer des enquêtes de satisfaction spécifiques pour comprendre les causes du mécontentement et prendre des mesures pour corriger la situation. Réparer le tord causé en proposant des réductions pour des achats à venir. Ces actions doivent intervenir rapidement.
- **Cluster 3 (9%)** : clients qui achetaient fréquemment, pour des montants peu élevés, et qui ne l'ont pas fait depuis un moment. Leur niveau de satisfaction est bas. **Clients perdus**.  
➡ Ne pas déployer de moyens pour maintenir ces clients (ils n'ont pas commandé depuis longtemps et dépensent peu). Analyser leurs retours pour comprendre les problèmes rencontrés par ces clients et ainsi éviter de les réitérer chez les clients des autres segments.
- **Cluster 4 (20%)** : clients qui dépensent peu, mais relativement fréquemment. Ils sont plutôt satisfaits de leur expérience. **Client à relancer/booster**  
➡ Communiquer de façon régulière avec ces clients pour ne pas se faire oublier. Mettre en place des programmes de fidélité ou des promotions ciblées pour relancer leurs envies d'achat.

# Stabilité temporelle

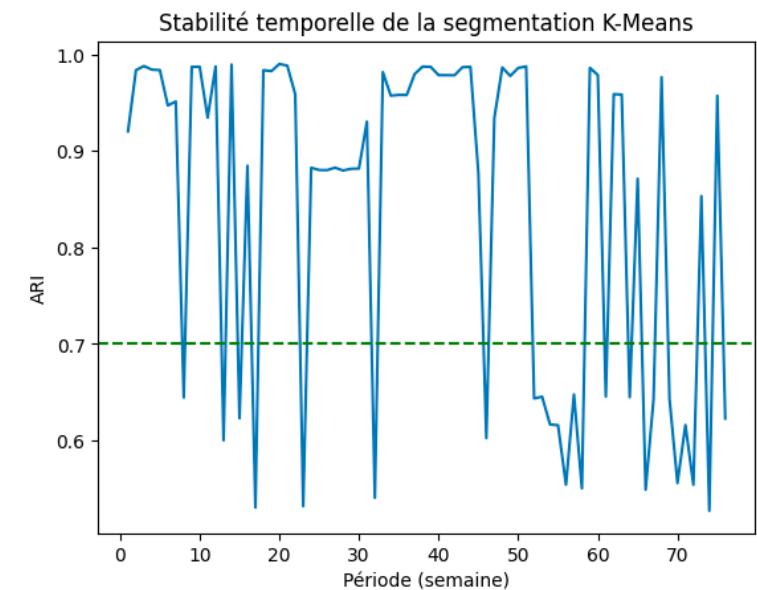
## Définition du délai de maintenance du modèle

- Le dataset couvre 76 semaines à partir du 15 septembre 2016
- Cohorte de client sur les 6 premiers mois, ce qui correspond à 4731 commandes (0,04% du dataset)

Méthodologie :



- L'ARI (indice de Rand ajusté) mesure la similarité entre deux segmentation: plus l'indice est élevé, plus les segmentations sont proches.
  - Indice = 1 : accord parfait entre les segmentations
  - Indice = 0 : accord au hasard
  - Indice = -1 : les segmentations sont complètement différentes



**Proposition d'un contrat de maintenance :**  
**mise à jour toutes les 3 semaines\***  
(période moyenne de segmentation avec un ARI > 0.7)

\* période moyenne de 3,7 semaines, arrondie à la baisse par prudence

# Limites des résultats obtenus

- La segmentation obtenue est dépendante des données sous-jacentes. Au cours de nos analyses nous avons noté **un certain nombre d'incohérences**, de nature humaine ou technologique, qui mériteraient d'être corrigées:
  1. Mise à jour de la Database géolocalisation
    - incohérence entre les données géographiques de clients (Europe et Asie) et leur ville, Etat et Zipcode brésiliens
    - il existe des zipcode dans la base clients et dans la base vendeurs qui ne sont pas répertoriés dans la base géolocalisation (le zip code est l'unique clé commune entre ces trois datasets)
  2. Vérification des Timestamps pour corriger ou investiguer les incohérences suivantes (process à revoir ?) :
    - dans 1359 cas, le transporteur se voit remettre une commande avant qu'elle soit approuvée (durée moyenne de 1 jour de retard)
    - dans 23 cas, le client reçoit sa commande avant qu'elle soit remise au transporteur (durée moyenne de 3,3 jour de retard)
  3. Rationalisation des caractéristiques produit
    - 610 produits n'ont pas de nom, description, photo
    - 2 produits n'ont pas de mesures et de poids
    - 4 produits ont un poids nul
  4. Revue des process de mise à jour des différentes databases
    - Nous avons identifié une commande qui n'était pas dans le dataset relatifs aux paiements. Nous n'avions donc pas le détail relatif aux type de paiement, facilité de paiement... relatif à cette commande
  5. Revue de l'exhaustivité / résilience des enregistrements
    - Aucune commande n'a été enregistrée entre octobre 2016 à janvier 2017
- Les résultats sont par ailleurs **limités par la profondeur des données communiquées** (23 mois) et **le type de clients analysés** (seuls 3% ont fait plus d'une commande).
- Enfin, une **segmentation plus fine pourrait être obtenue à partir de données personnelles** du client telles que l'âge, le sexe, la catégorie socio-professionnelle, la situation familiale, les hobbies... Ces données devront être récoltées, traitées et stockées en adéquation avec le RGPD.

# Annexes

# Technologies utilisées

- Les travaux ont été réalisés sur Python (3.9.6) avec les bibliothèques ci-dessous:
  - numpy 1.23.5
  - pandas 2.0.2
  - geopandas 0.13.2
  - matplotlib 3.7.1
  - seaborn 0.12.2
  - plotly 5.15.0
  - scikit-learn 1.2.2
  - yellowbrick 1.5
- Les modèles de machine learning ont été implémentés (et leurs scores calculés) grâce à la librairie **scikit-learn** via les méthodes suivantes:
  - KMeans : sklearn.cluster.KMeans
  - Hierarchical clustering : sklearn.cluster.AgglomerativeClustering
  - DBSCAN : sklearn.cluster.DBSCAN
  - Scores : sklearn.metrics.silhouette\_score / calinski\_harabasz\_score / davies\_bouldin\_score
  - ARI : sklearn.metrics.adjusted\_rand\_score
  - GridSearch : sklearn.model\_selection.GridSearchCV