Segmenter les clients du site d'e-commerce

olist store

Contexte

olist store propose une **solution de vente** sur les marketplaces en ligne du Brésil.

Cette solution permet aux **petits commerçants** de vendre leurs produits en ligne facilement.

Mais pour les aider et améliorer son chiffre d'affaires, **olist** doit leur **fournir les moyens d'adapter leurs offres** à la clientèle du site.

Objectif du projet

olist store

L'entreprise souhaite que l'on fournisse à ses équipes d'e-commerce une segmentation des clients qu'elles pourront utiliser au quotidien pour leurs campagnes de communication.

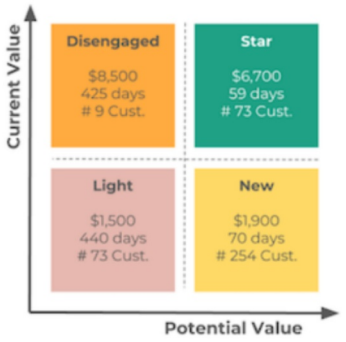
Il nous faut donc comprendre les données et les utiliser au mieux pour identifier différents types de consommateurs à l'aide d'algorithmes de segmentation (clustering).

Enfin, il faudra conduire une analyse de la stabilité du modèle pour proposer un contrat de maintenance adapté.

Segmentation de données

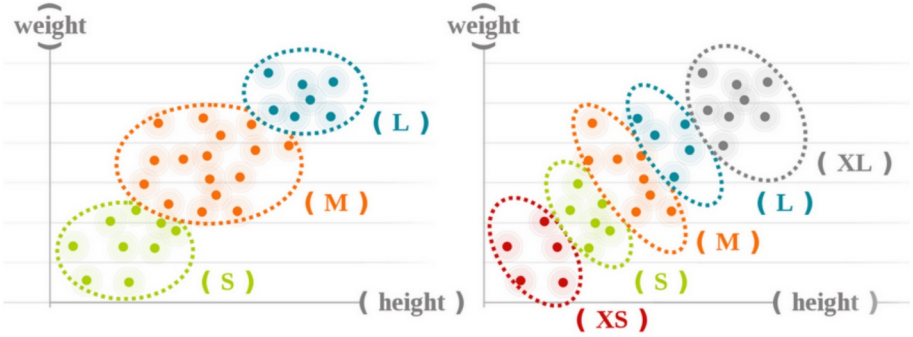
olist store

Dans le cadre du marketing, segmenter consiste à créer des groupes homogènes de clients que l'on peut ensuite activer dans des campagnes ou produits ciblés.



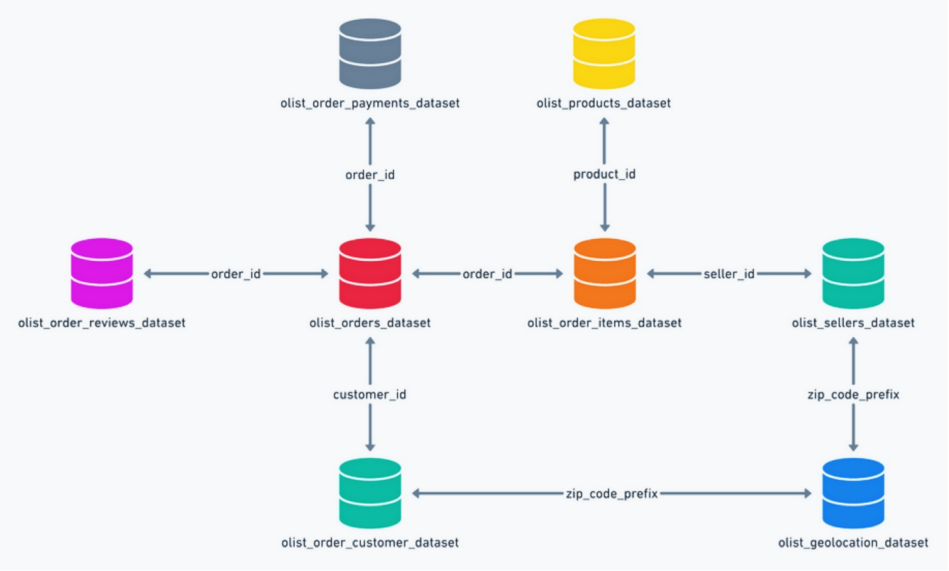
Segmentation de données

Plus généralement, une segmentation consiste à trouver des similitudes dans les échantillons de données pour identifier des groupes inconnus ou les découper selon un besoin précis.



Présentation du jeu de données

Il existe 9 fichiers source décrivant clients, commerçants, produits, commandes ou avis par 45 variables (après jointures).



Présentation du jeu de données

Il existe 9 fichiers source décrivant clients, commerçants, produits, commandes ou avis par 45 variables (après jointures).

- 99441 commandes uniques
- 96096 clients uniques
- 32951 références produits
- 3095 commerçants

Analyse exploratoire

olist store

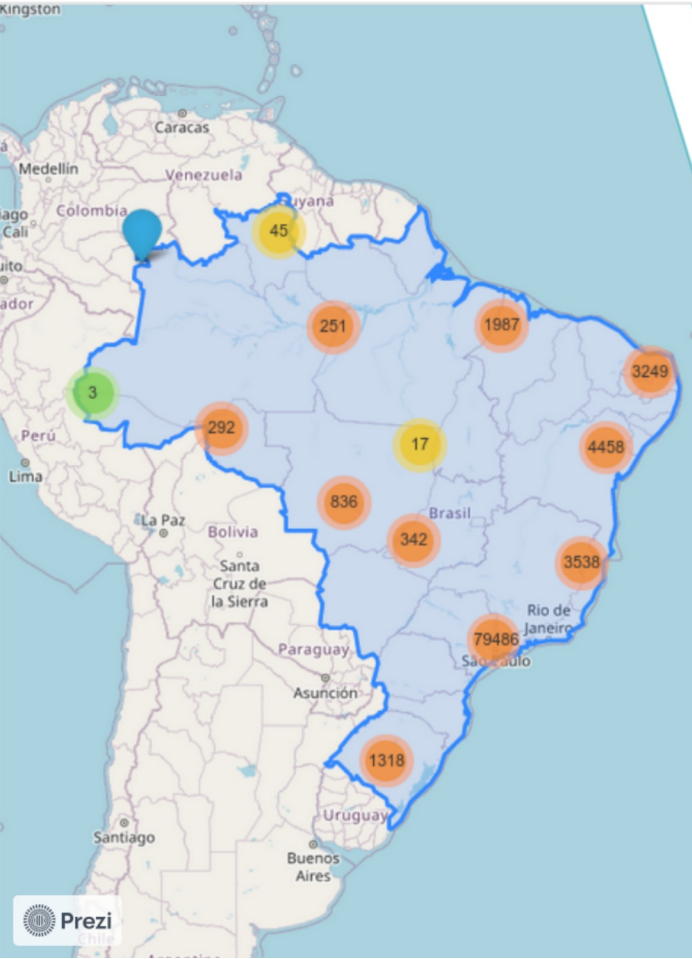
Les vérifications des erreurs les plus fréquentes
(valeurs manquantes, doublons, outliers, erreurs de format, erreurs lexicales, contenus multiples)
puis les analyses univariées et multivariées
ont permis de déceler quelques problèmes qui
ont été corrigés avec les actions suivantes:

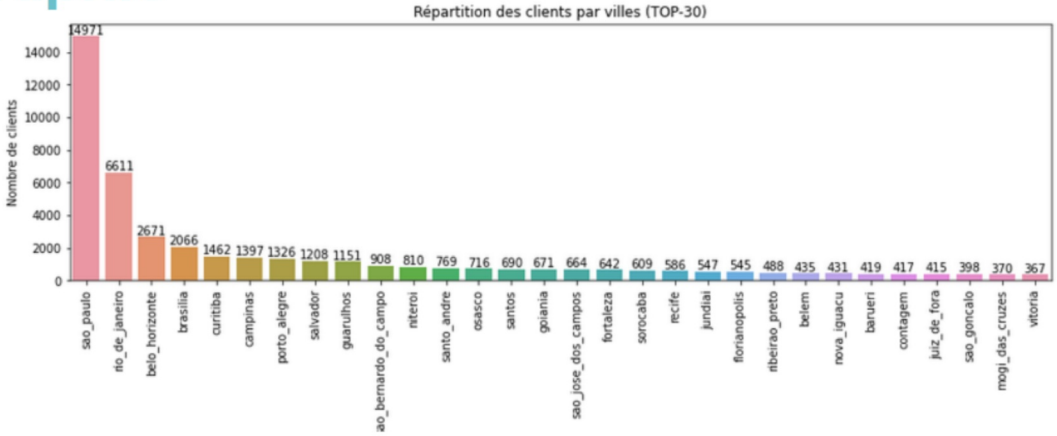
- Suppression des colonnes inutiles (jointures)
- Traitement des doublons
- Traitement des outliers
- Normalisation des données

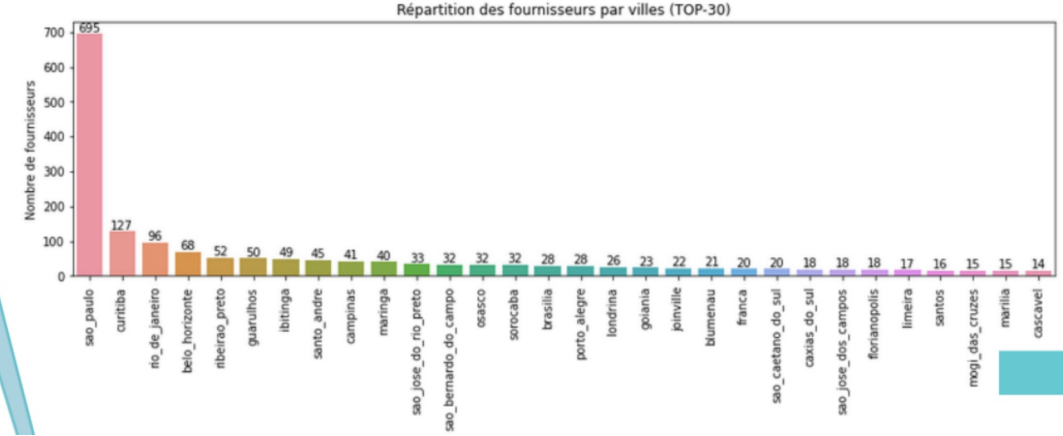
Analyse exploratoire

olist store

Géographie



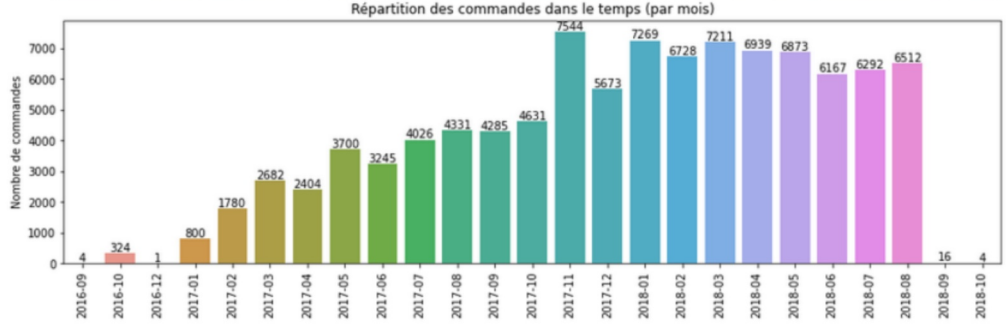




Analyse exploratoire

olist store

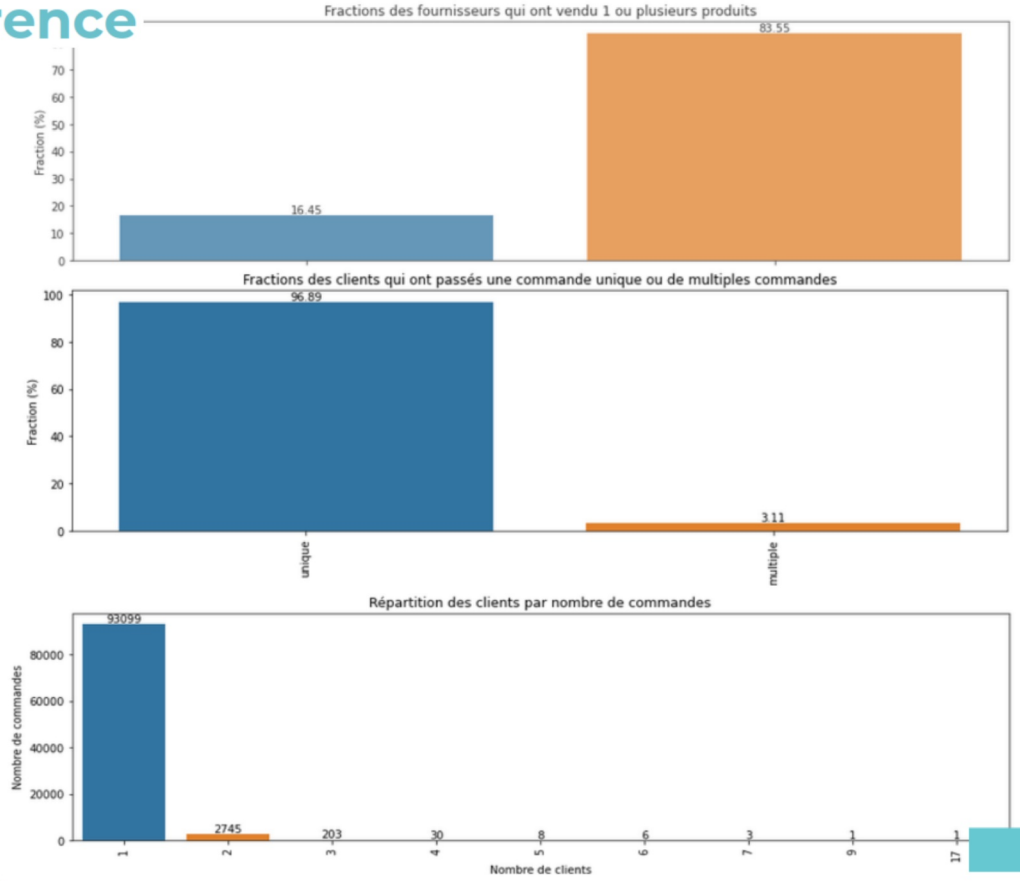
Temporalité



Analyse exploratoire

olist store

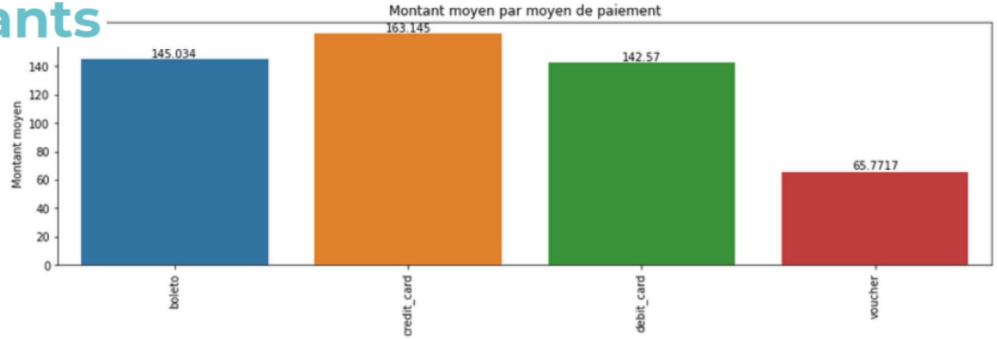
Récurrence



Analyse exploratoire

olist store

Montants

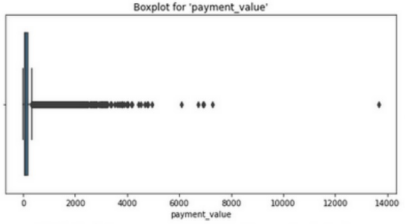


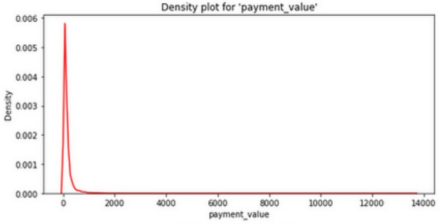
----- UNIVARIATE ANALYSIS for "payment_value" -----

count	103886.000000
mean	154.100380
std	217.494064
min	0.000000
25%	56.790000
50%	100.000000
75%	171.837500
max	13664.080000

----- UNIVARIATE ANALYSIS for "price" -----

count	112650.000000
mean	120.653739
std	183.633928
min	0.850000
25%	39.900000
50%	74.990000
75%	134.900000
max	6735.000000

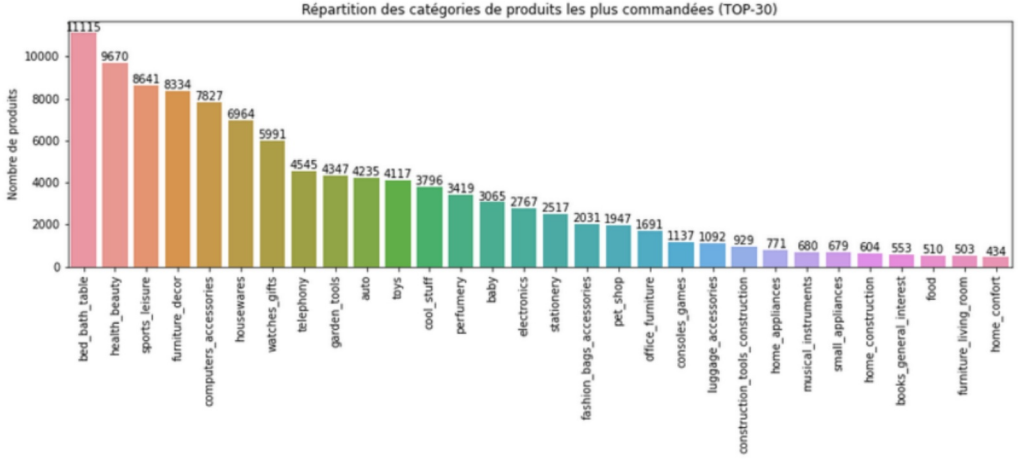


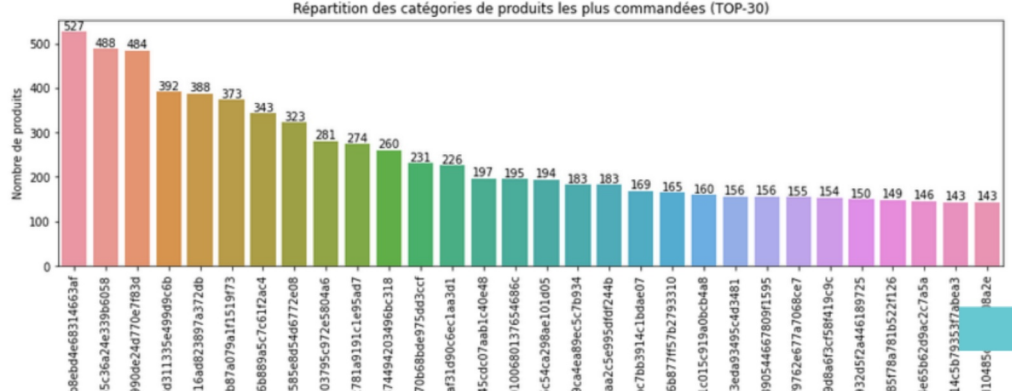


Analyse exploratoire

olist store

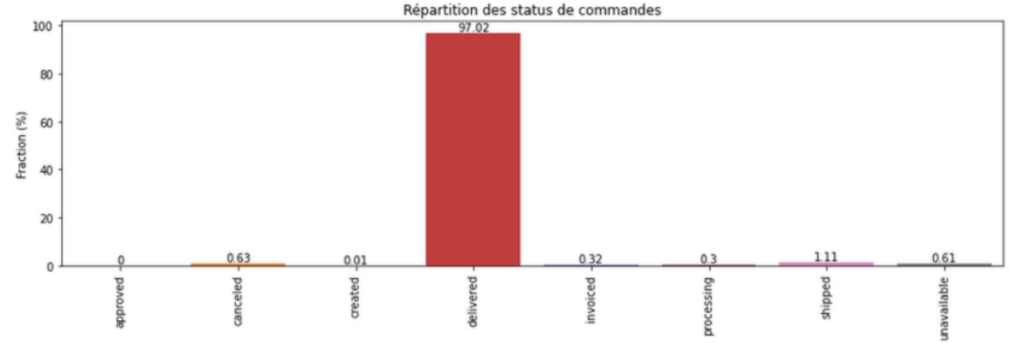
Catégories

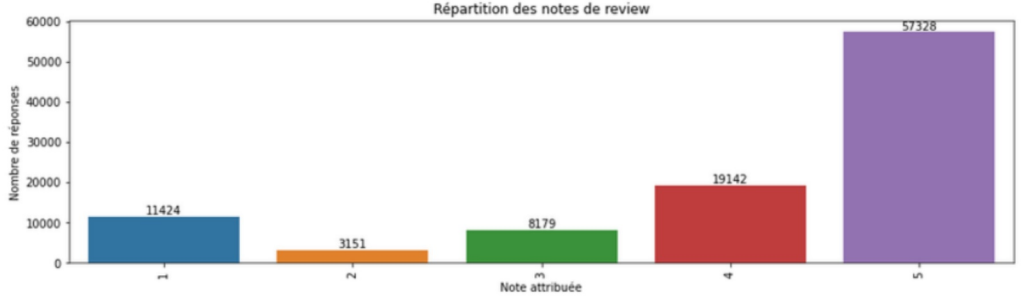




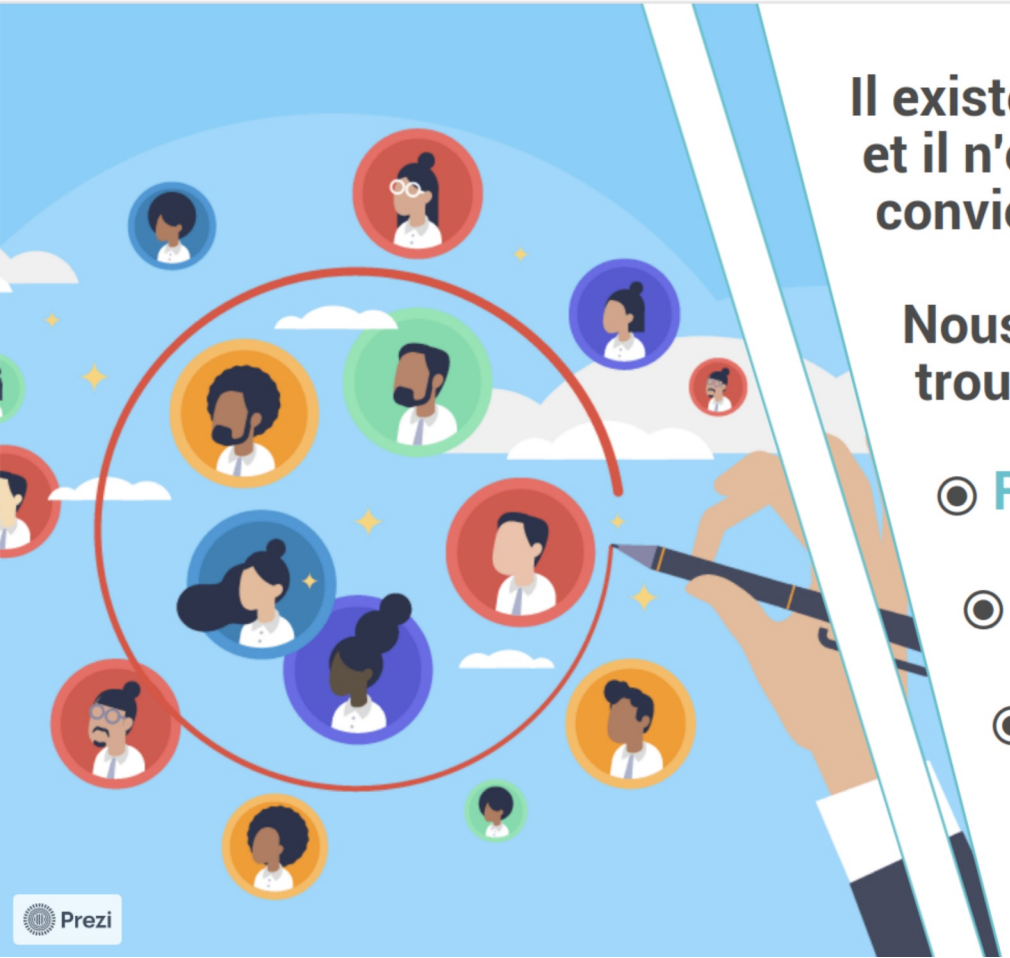
Analyse exploratoire

Suivi





Recherche des segments



Il existe de nombreux algorithmes de ML, et il n'est pas évident de savoir celui qui convient le mieux à un problème donné...

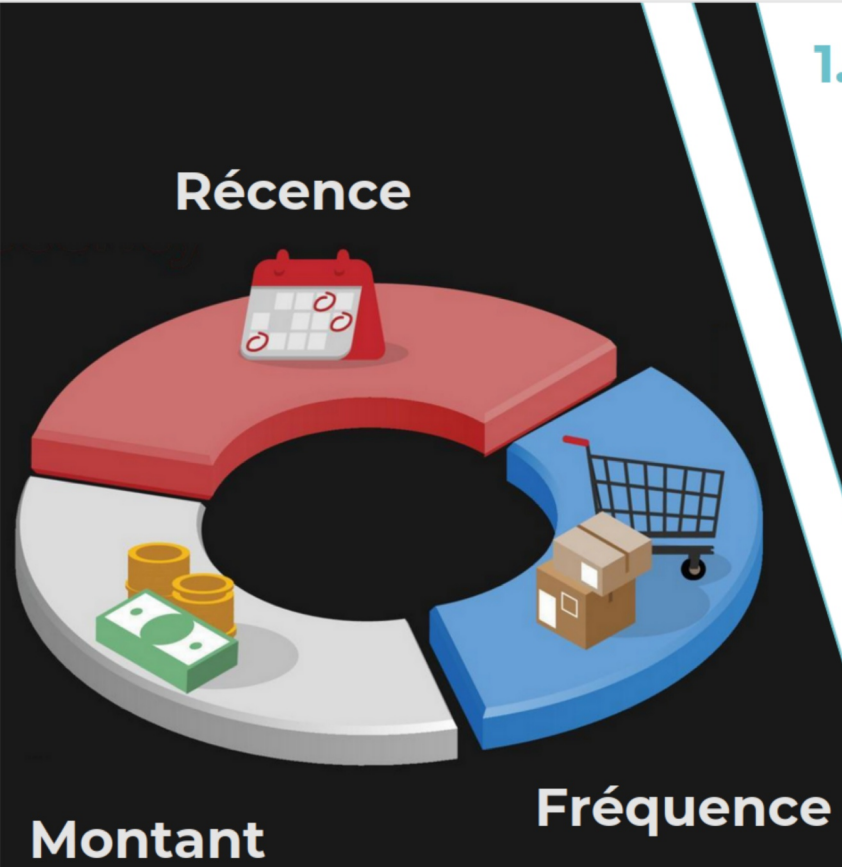
Nous allons donc en tester plusieurs pour trouver celui qui nous convient le mieux.

- RFM
- KMeans
- Classification Ascendante Hiérarchique
- DBSCAN

Recherche des segments

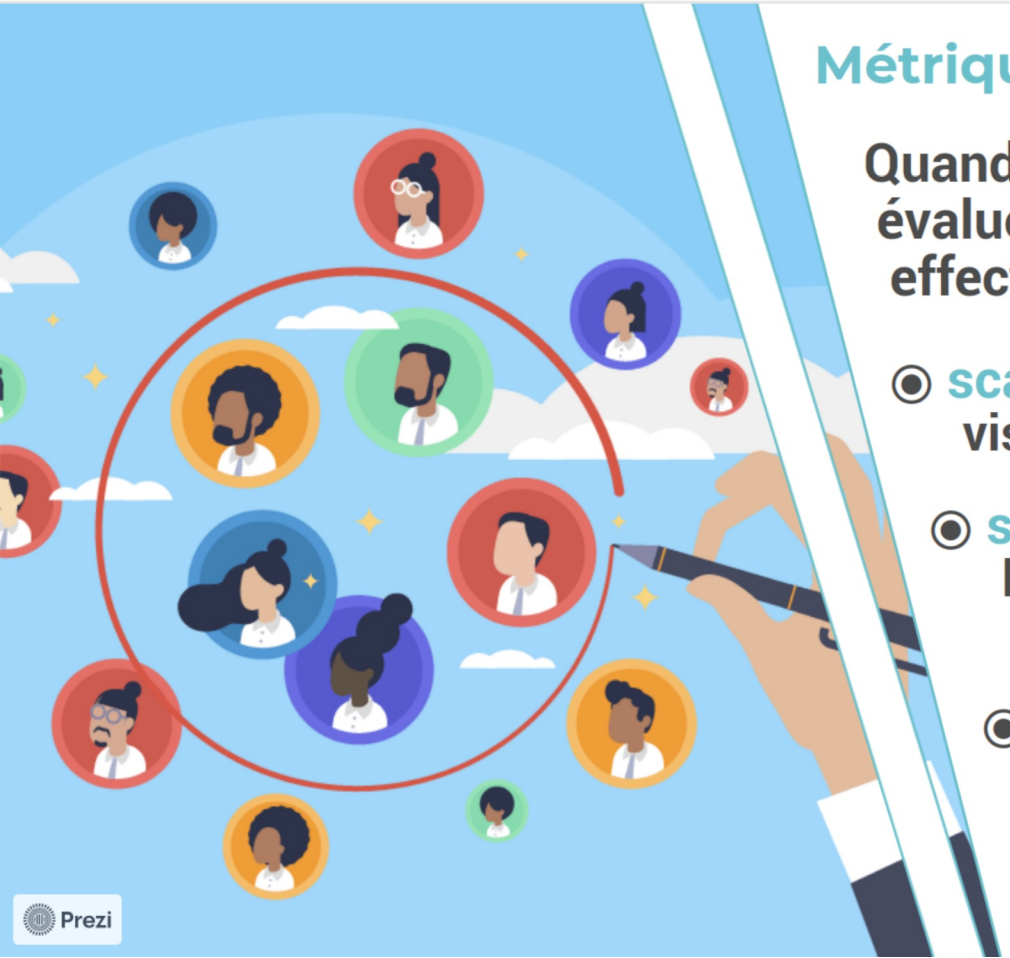
1. RFM (Récence | Fréquence | Montant)

```
score:1 | Recence:[ -1000;-229.0] | Frequence:[     0;   1.0] | Montant:[     0; 75.25]
score:2 | Recence:[-229.0;-122.0] | Frequence:[   1.0;   1.0] | Montant:[ 75.25;152.71]
score:3 | Recence:[-122.0;  -3.0] | Frequence:[   1.0;  11.0] | Montant:[152.71;7274.88]
```



score	recence	frequence	montant	count
111	-290.226064	1.000000	50.639854	8011
112	-289.737503	1.000000	109.486304	7962
113	-291.337739	1.000000	331.687329	7491
131	-293.400000	2.000000	63.639000	20
132	-297.367647	2.009804	118.064657	204
133	-298.699670	2.176568	379.324670	606
211	-176.487552	1.000000	49.803600	8114
212	-173.925050	1.000000	110.500962	7525
213	-174.264780	1.000000	320.303677	7561
231	-172.558824	2.000000	60.616471	34
232	-182.742857	2.014286	118.355929	140
233	-181.701195	2.097610	366.048008	502
311	-61.460323	1.000000	49.438063	7914
312	-62.951949	1.000000	110.265900	8158
313	-63.712323	1.000000	339.826873	7717
331	-81.000000	2.000000	57.104545	11
332	-70.986301	2.027397	119.180000	73
333	-70.759804	2.049020	433.589216	204

Recherche des segments



Métrique

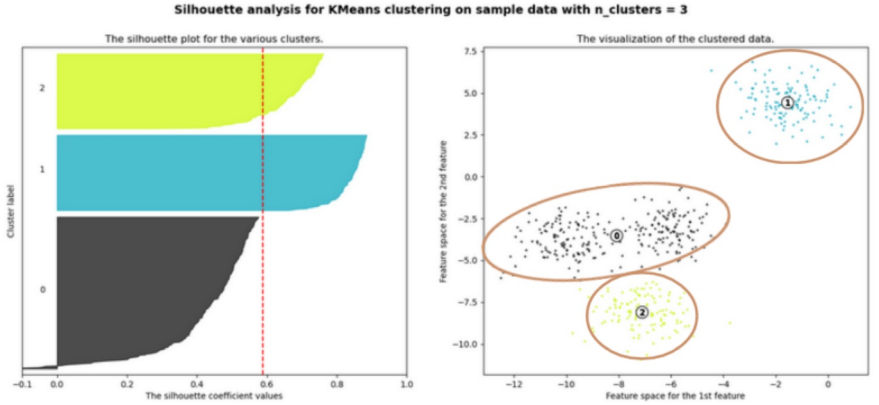
Quand on ne dispose pas d'une variable cible pour évaluer la qualité du modèle, l'évaluation doit être effectuée en utilisant le modèle lui-même.

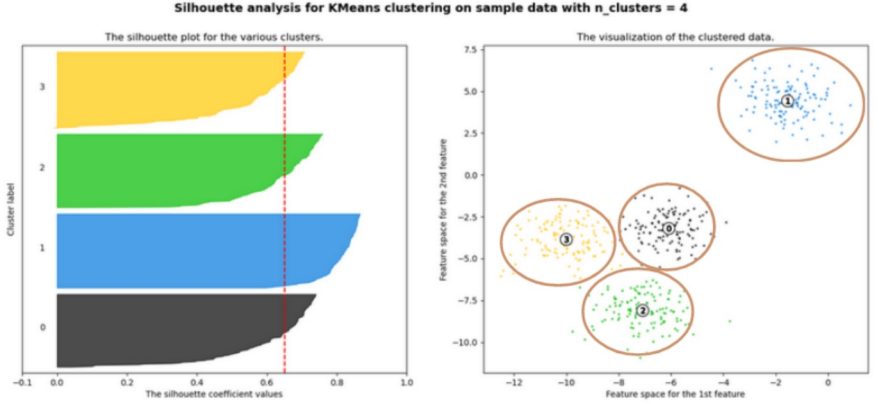
- scatter plot : le moyen le plus direct est de visualiser les segments quand c'est possible...
- score silhouette : tente d'indiquer si les échantillons ont été rattachés au "bon" cluster ou pas.
- david-bouldin score : qui est le ratio entre les écarts intra et inter-groupes, permet d'estimer le nombre de cluster idéal.

Recherche des segments

olist store

Silhouette & scatter plot



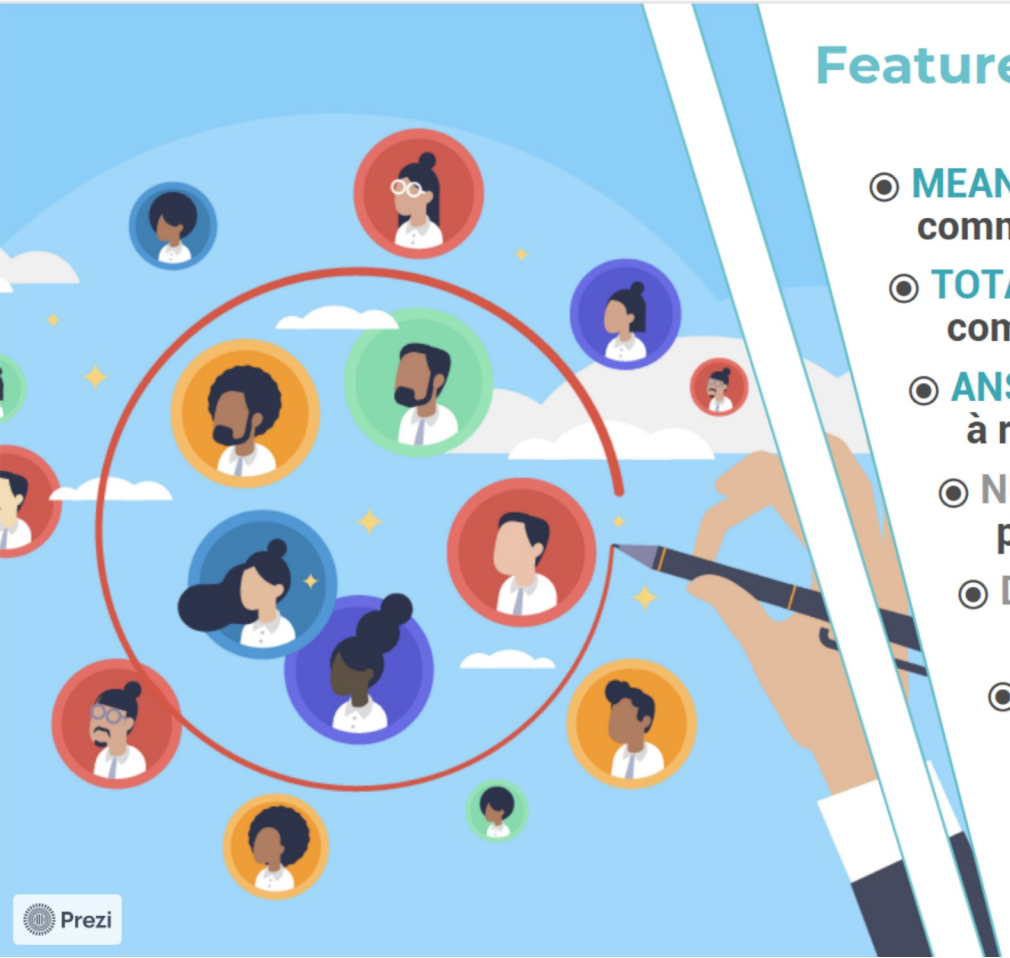


Recherche des segments

Feature Engineering

- **RECENCE & R_SCORE : A combien de jours remonte le dernier achat ?**
- **MONTANT & M_SCORE : Quel est le montant total des achats sur la période choisie ?**
- **REVIEW_SCORE : Quelle est la note moyenne attribuée par un client pour ses commandes ?**
- **GEOLOCATION_LAT : La latitude du client lors de la dernière commande.**
- **GEOLOCATION_LNG : La longitude du client lo de la dernière commande.**

Recherche des segments

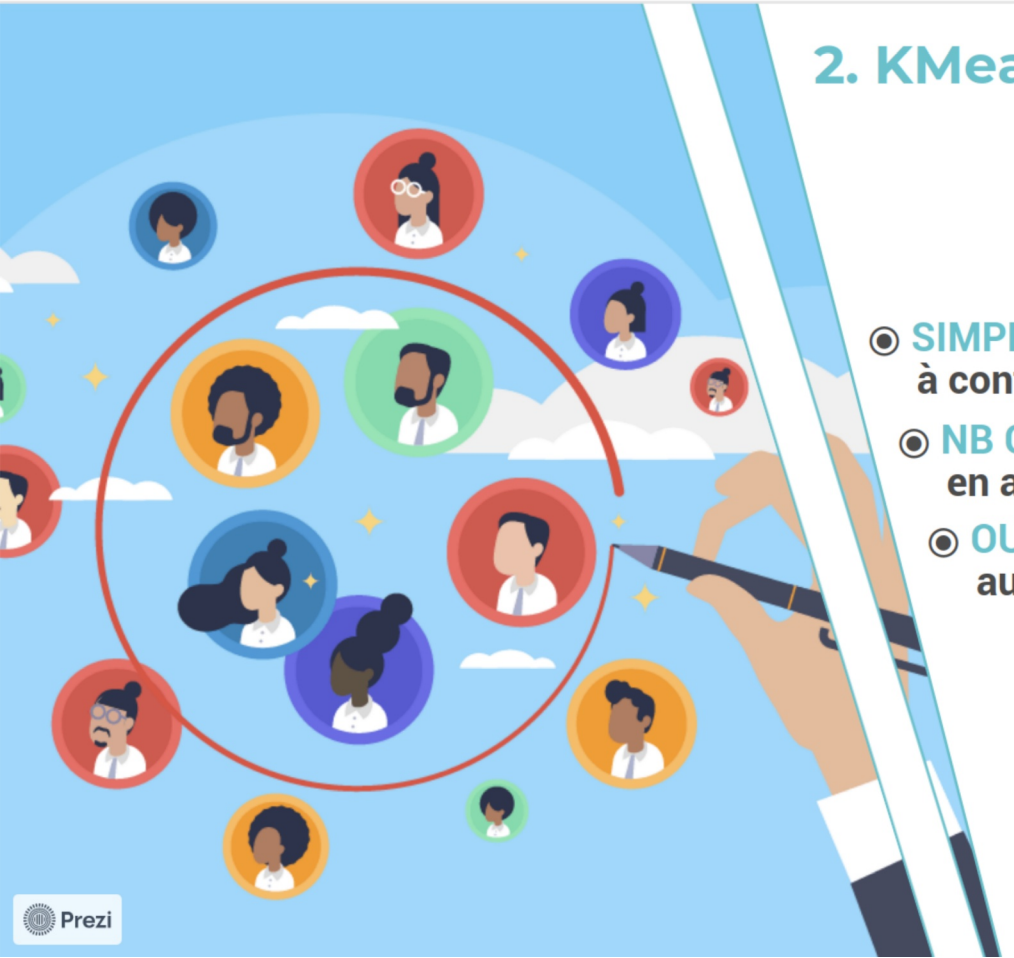


Feature Engineering

- **MEAN_ITEMS** : Quel est le nombre moyen de produits commandés par le client ?
- **TOTAL_ITEMS** : Quel est le nombre total de produits commandés sur la période choisie ?
- **ANSWER_DAYS** : Combien de jours à mis le client à répondre à l'énquête de satisfaction ?
- **NUM_ORDERS** : Combien de commandes ont été passées sur la période choisie ?
- **DELIVERY_DAYS** : Combien de jours ont été nécessaires en moyenne pour livrer ce client ?
- **AVG_NUM_COMMENTS** : Quelle est le taux moyen de commentaires laissés par ce client ?
- **AVG_NUM_COMMENTS** : Quelle est le taux moyen de commentaires laissés par ce client ?

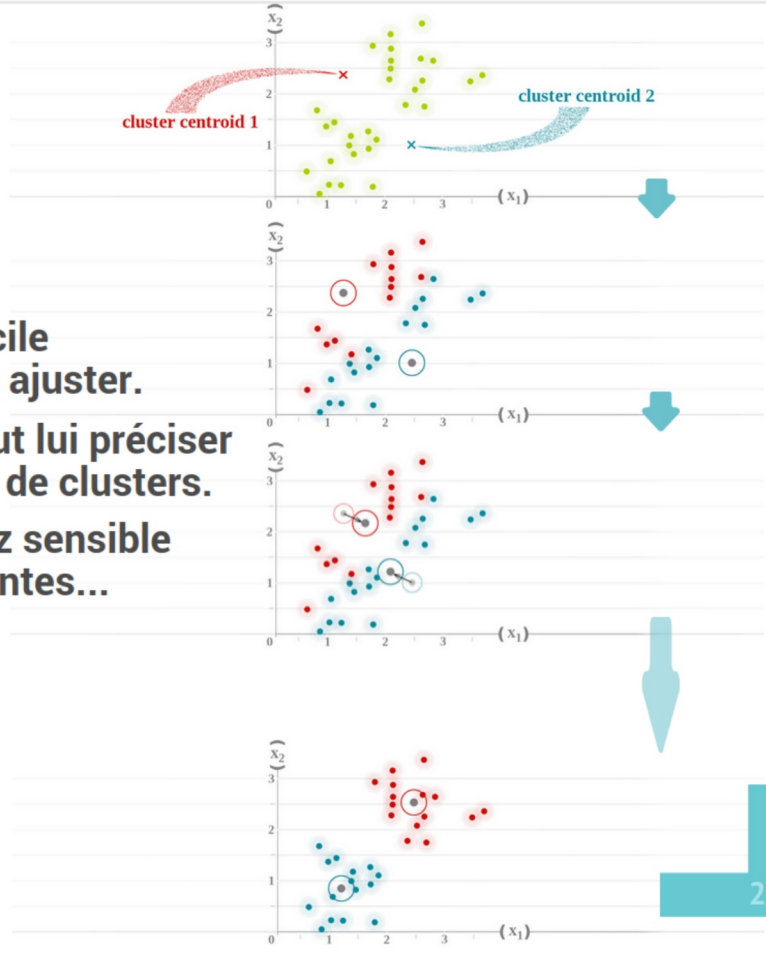
Recherche des segments

olist store



2. KMeans

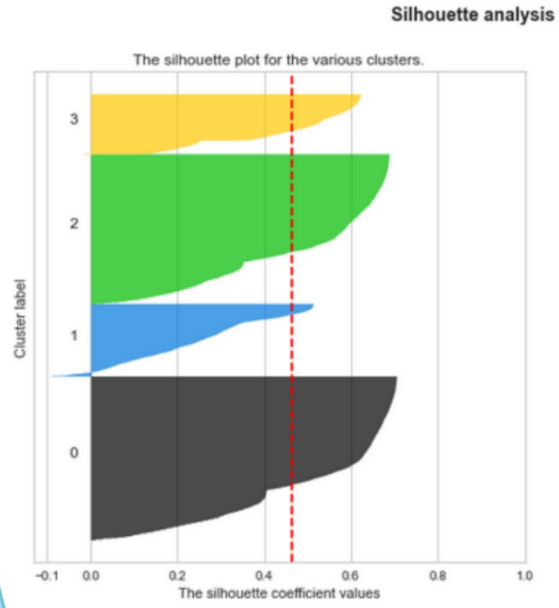
- **SIMPLE** : il est très facile à configurer et donc à ajuster.
- **NB CLUSTERS** : il faut lui préciser en amont le nombre de clusters.
- **OUTLIERS** : il assez sensible aux valeurs aberrantes...

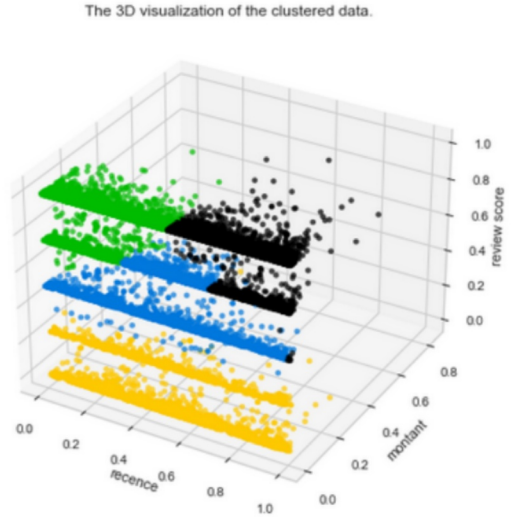


Recherche des segments

2. KMeans

```
Silhouette score moyen: 0.463
```

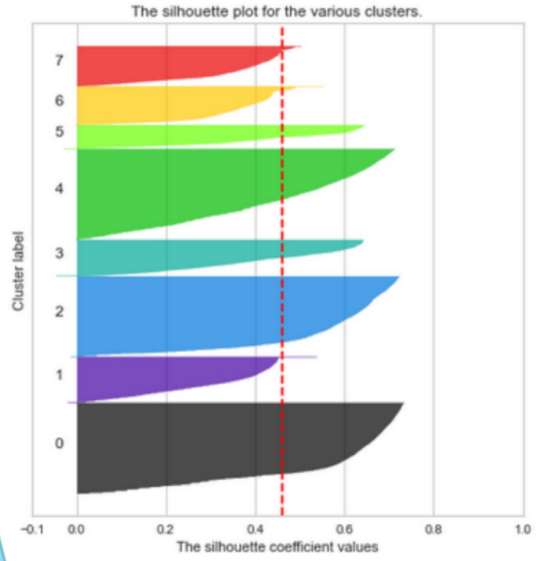


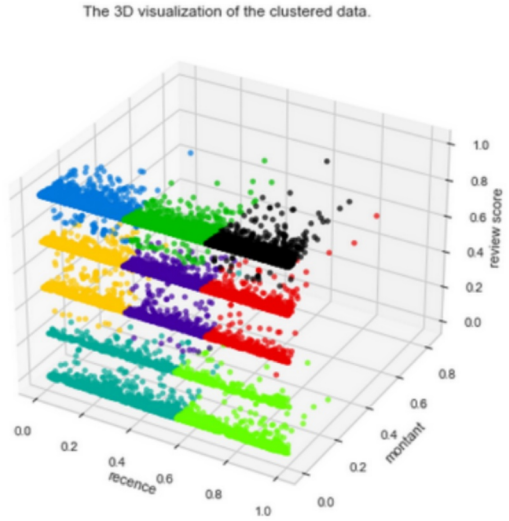


Recherche des segments

2. KMeans

```
Silhouette score moyen: 0.459
```

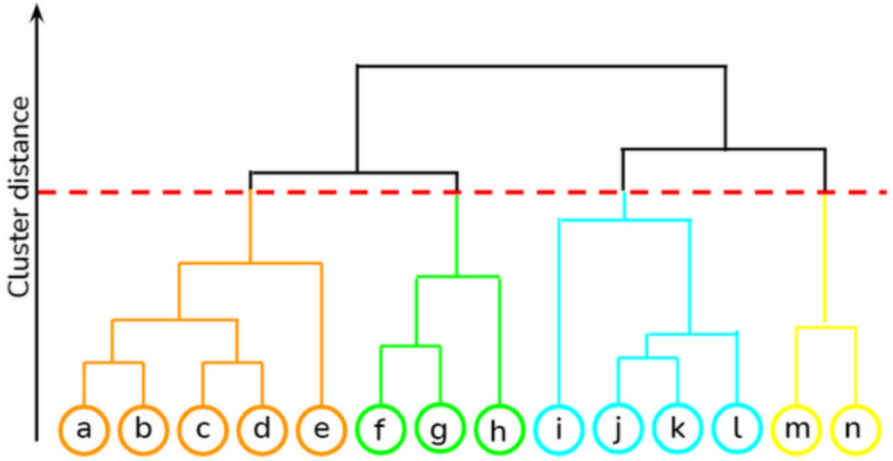




Recherche des segments

3. Classification Ascendante Hiérarchique

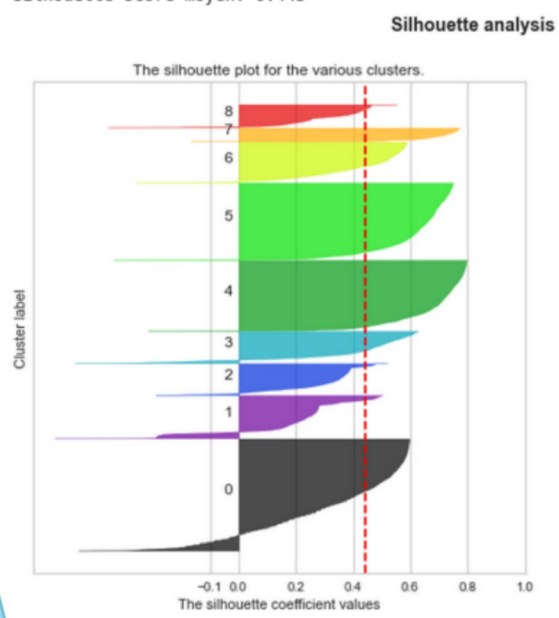
- **RAPIDE** : il n'est pas nécessaire de l'entraîner plusieurs fois pour essayer différentes segmentations.
- **NB CLUSTERS** : il n'est pas nécessaire de lui indiquer le nombre de clusters en amont.
- **VISUEL** : on peut facilement voir les clusters potentiels avec un Dendrogramme.

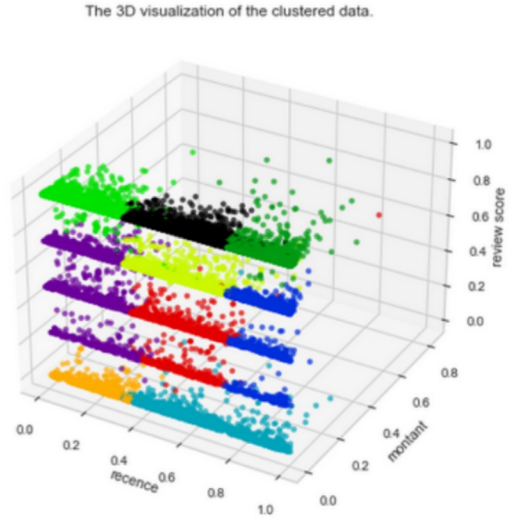


Recherche des segments

3. Classification Ascendante Hiérarchique

```
Silhouette score moyen: 0.443
```

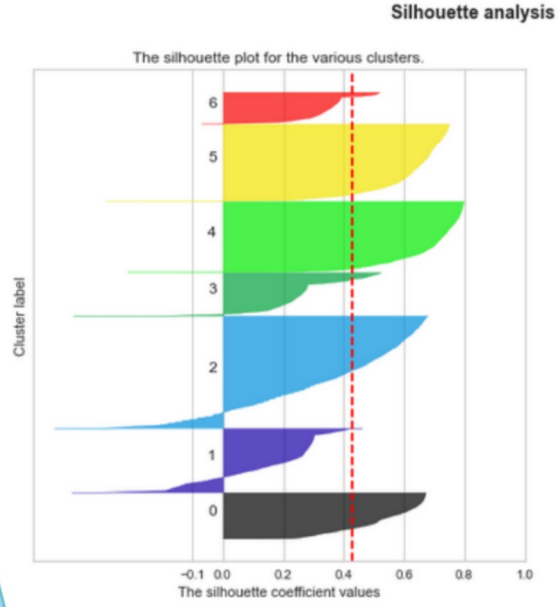


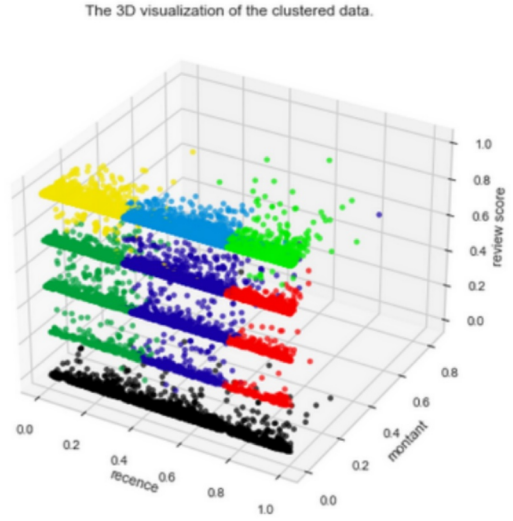


Recherche des segments

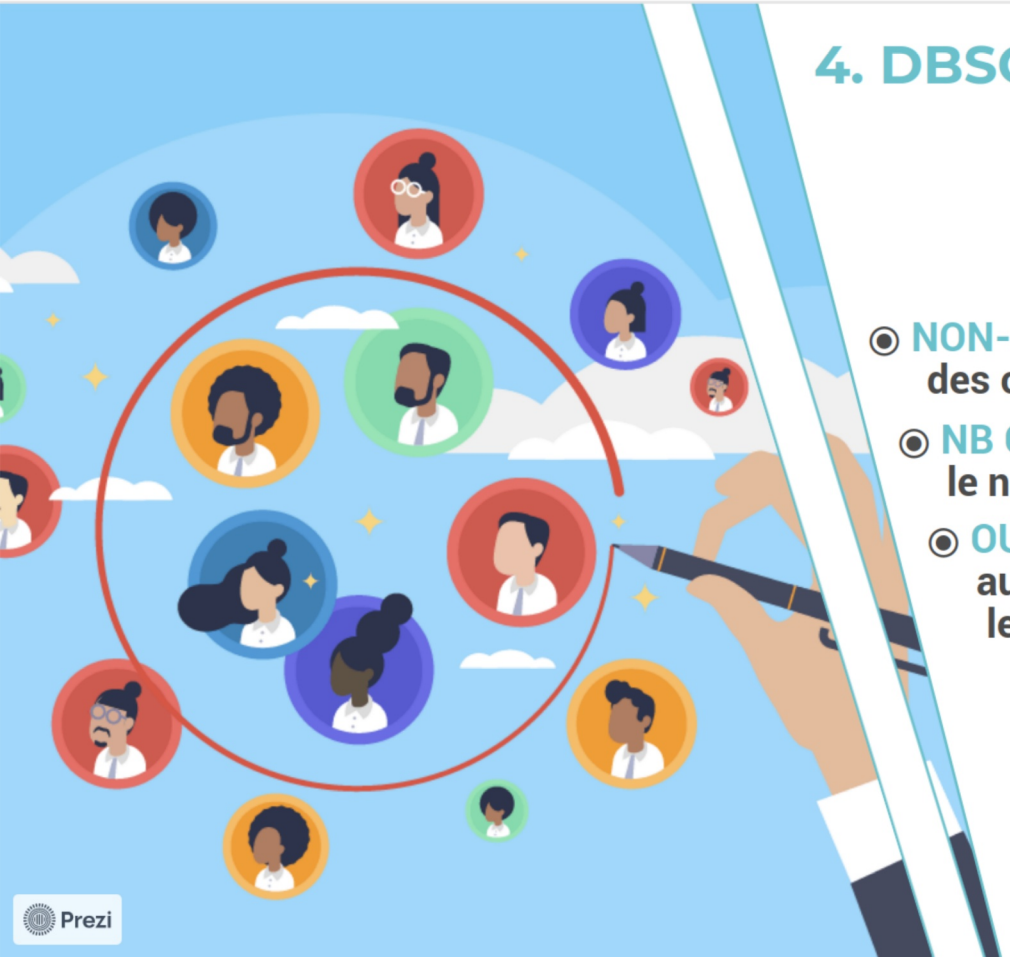
3. Classification Ascendante Hiérarchique

```
Silhouette score moyen: 0.430
```



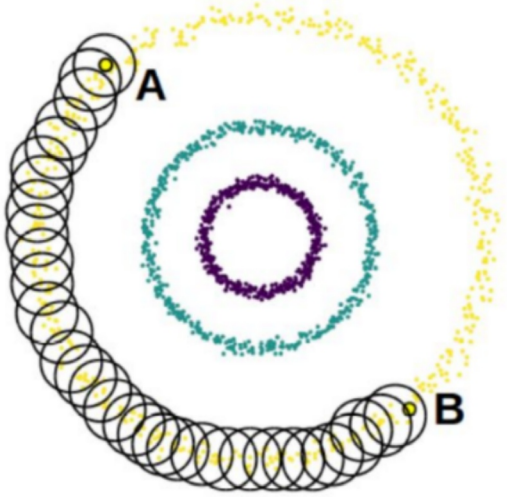


Recherche des segments

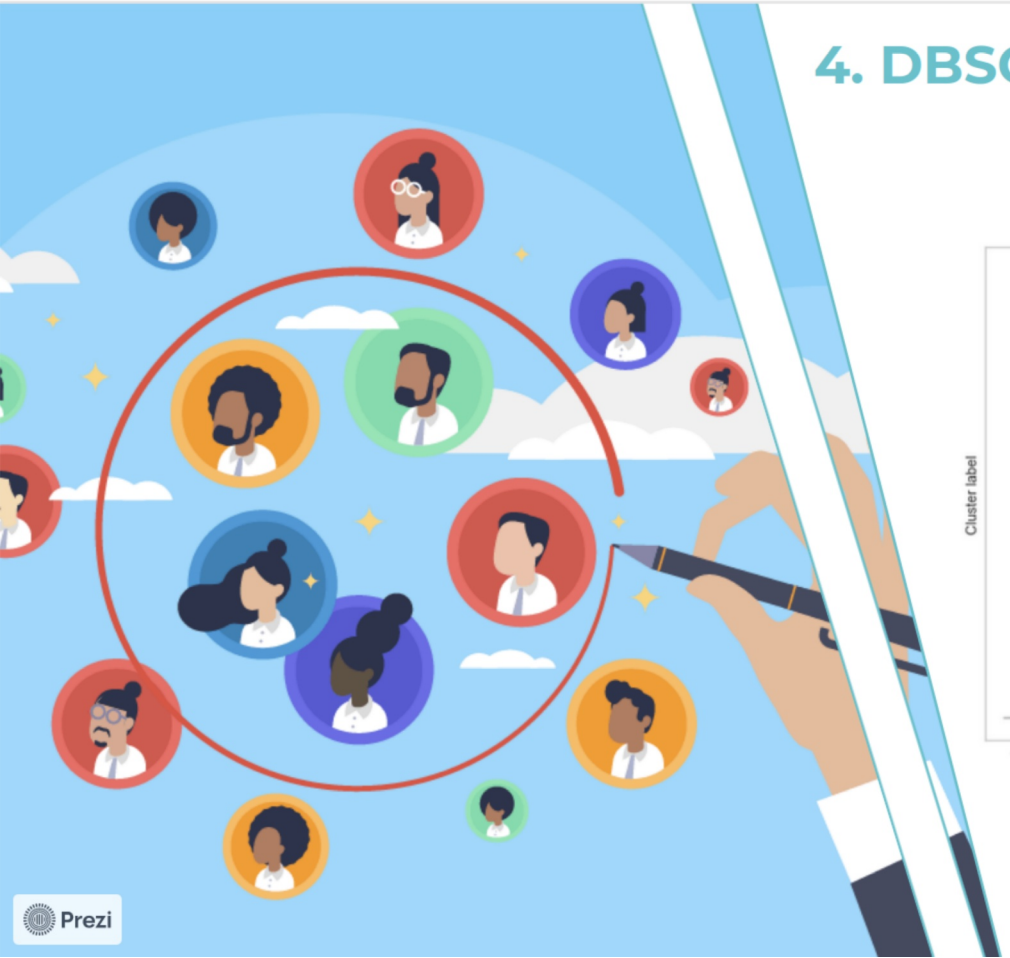


4. DBSCAN

- **NON-CONVEXE** : il peut trouver des clusters de forme arbitraire.
- **NB CLUSTERS** : il trouve seul le nombre de clusters.
- **OUTLIERS** : il est résistant aux outliers et permet de les identifiers comme BRUIT.

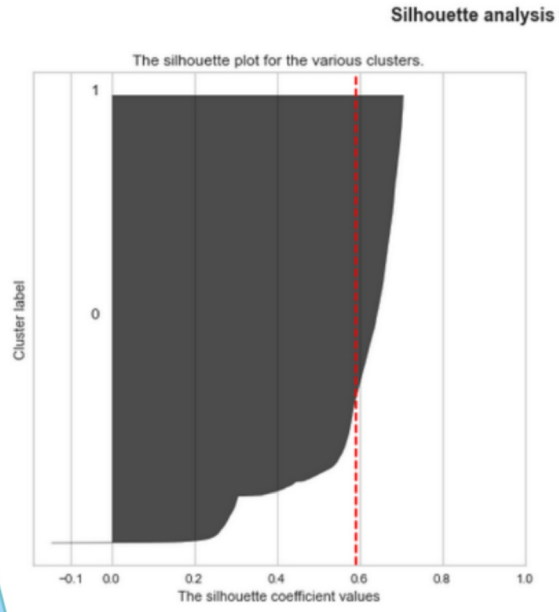


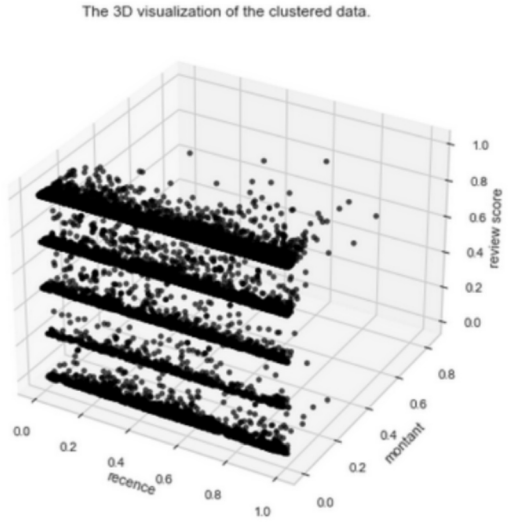
Recherche des segments



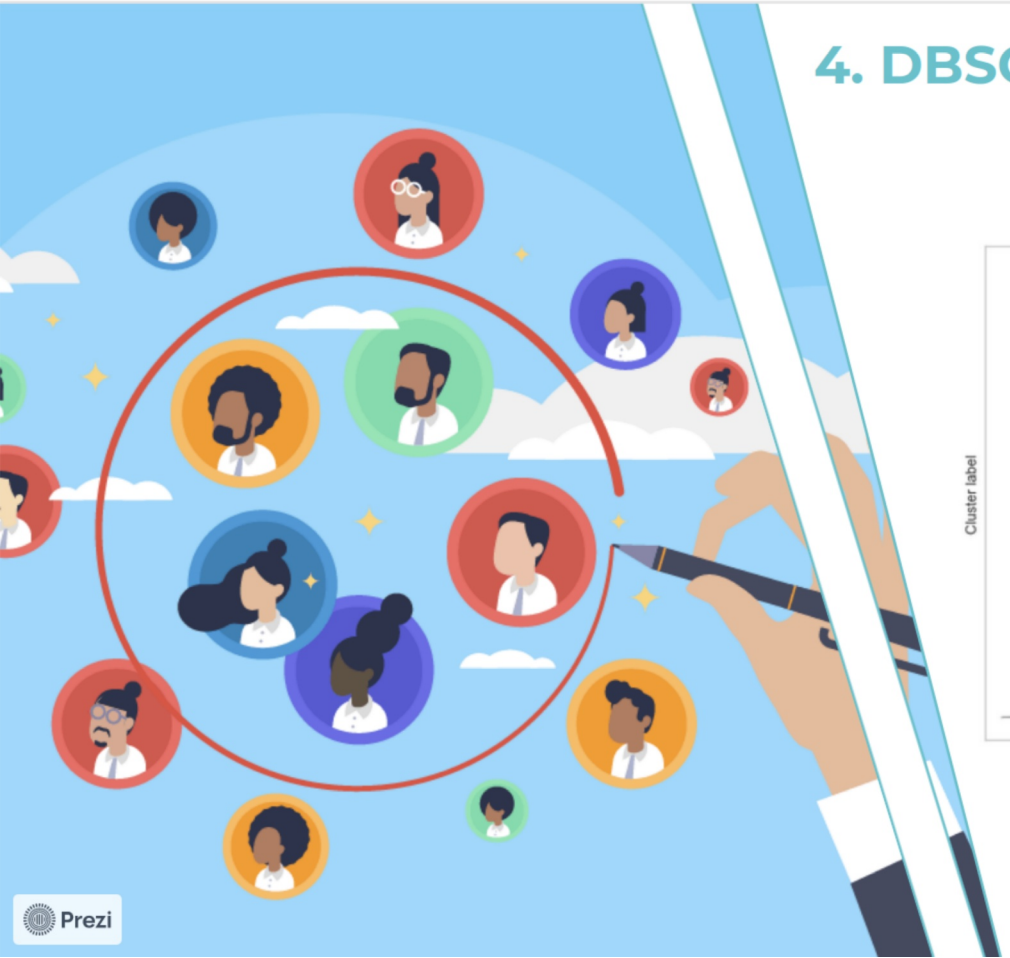
4. DBSCAN

SILHOUETTE : 0.5925



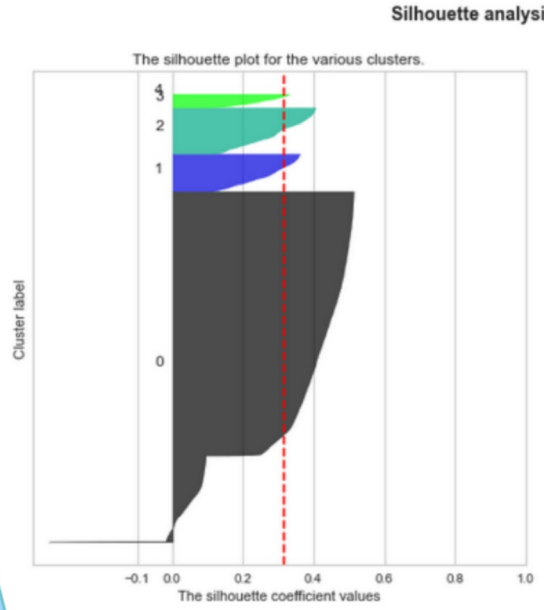


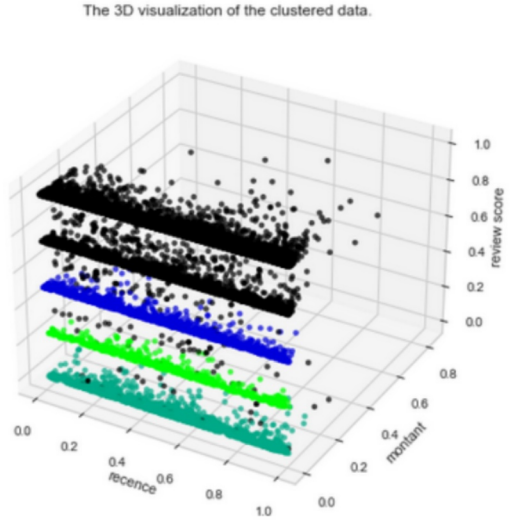
Recherche des segments



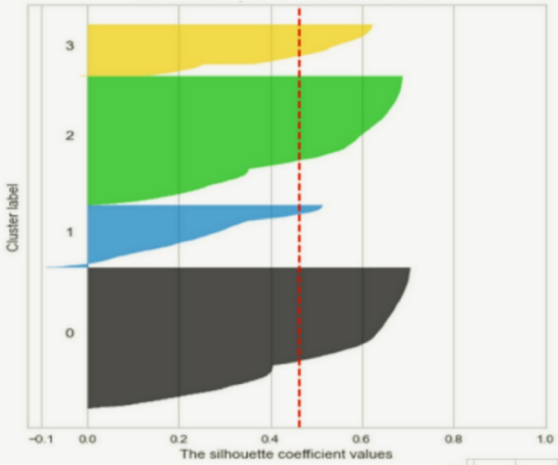
4. DBSCAN

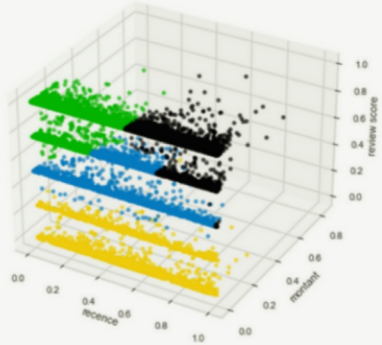
SILHOUETTE : 0.3174

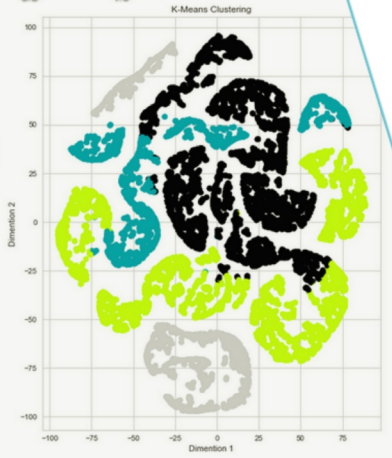




Modèles finaux







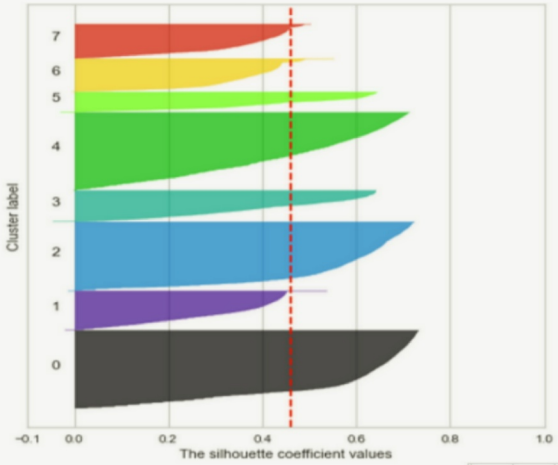
1. KMeans avec 4 clusters (0.463)

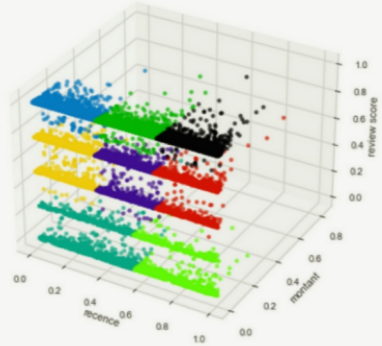
(meilleurs score silhouette)

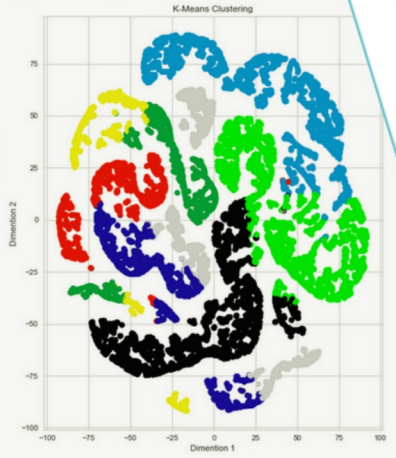
	recence	montant	review_score	mean_items	total_items	
2	-272.099658	162.018386	4.829357	1.104567	1.206604	les clients satisfaits et que l'on a PAS vu récemment
1	-186.997535	169.650065	3.468423	1.144794	1.268022	les clients moyens en tout
3	-184.441878	196.693152	1.228204	1.319672	1.390089	les clients dépensiers mais insatisfaits
0	-85.324622	166.391401	4.828683	1.111542	1.155846	les clients satisfaits et que l'on a vu récemment

Modèles finaux

olist store



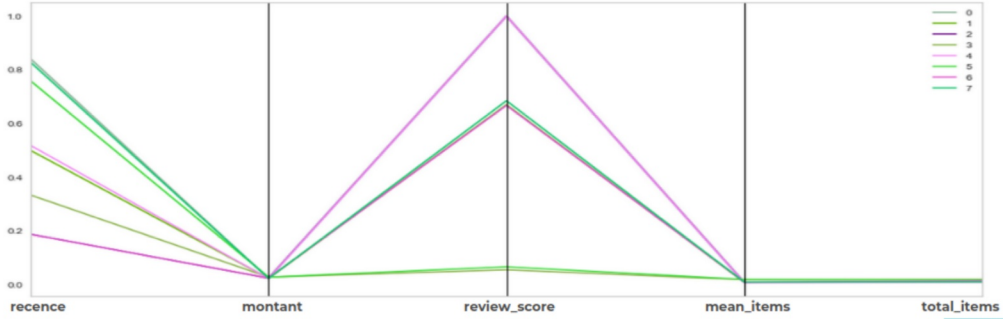




2. KMeans avec 8 clusters (0.459)

(semble plus intéressant)

	recence	montant	review_score	mean_items	total_items	
2	-298.081389	169.581064	4.993505	1.112595	1.236111	les clients satisfaits et inactifs
6	-297.970046	172.941682	3.662702	1.137673	1.275922	les clients modérément satisfaits et inactifs
3	-244.997543	197.351345	1.212633	1.311323	1.409705	les clients insatisfaits mais dépensiers et inactifs
1	-184.824732	165.497371	3.673418	1.126420	1.228335	les clients modérément satisfaits et peu actifs
4	-178.367282	159.027424	4.994875	1.096351	1.164576	les clients satisfaits et peu actifs
5	-91.098113	196.311434	1.257075	1.331604	1.361321	les clients insatisfaits mais dépensiers et actifs
7	-66.029067	163.494036	3.732439	1.147010	1.192286	les clients modérément satisfaits mais actifs
0	-61.365290	165.036910	4.997706	1.106563	1.145034	les clients satisfaits et actifs



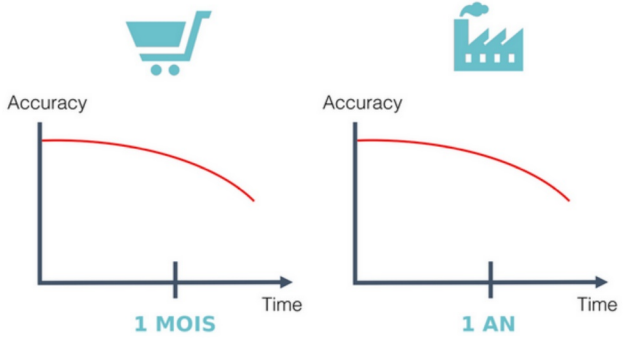
Détérioration du modèle

Problèmatique

Les modèles ont une tendance à devenir moins performant avec le temps...

Même si rien de radical ne se produit, des petits changements peuvent survenir et s'accumuler

(les produits, la clientèle, les marchés et l'économie en générale changent, et même sur un projet plus industriel, les capteurs ou les matières évoluent)



Détérioration du modèle

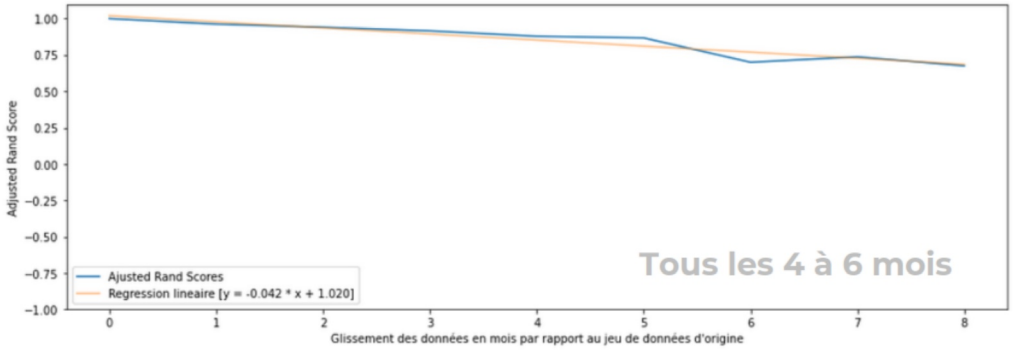
Métrique

Il existe de nombreux moyens d'évaluer la qualité d'une segmentation (Silhouette score, Calinski-Harabas, Davies-Bouldin etc.) mais si l'on veut suivre son évolution le choix le plus approprié semble être l'ARI ou l'AMI car ils ne sont pas affectés par les permutations de clusters et sont insensible à l'algorithme choisi.

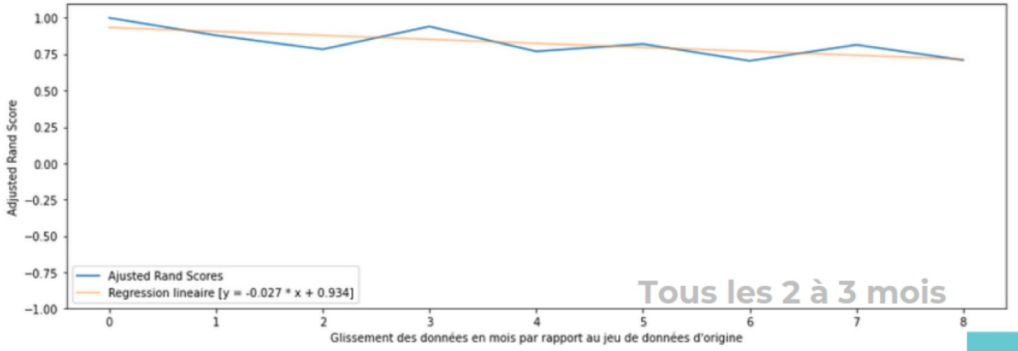
- Adjusted Rand Index : est adapté lorsque les clusters sont de taille vaguement similaires
- Adjusted Mutual Information : est adapté en cas de clusters nettement déséquilibrés

Détérioration du modèle

Évolution du modèle KMeans avec 4 clusters



Évolution du modèle KMeans avec 8 clusters

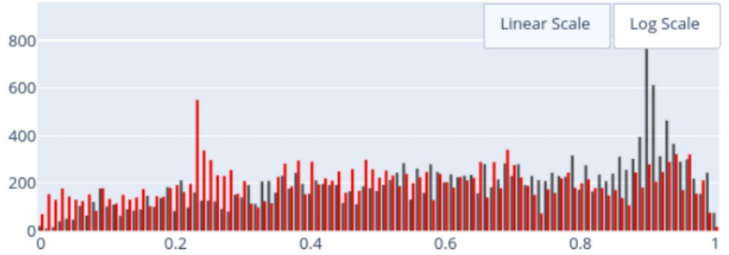


Détérioration du modèle

olist store

Évolution de la distribution des features

recence	reference	current
count	20000	20000
mean	0.61	0.52
std	0.27	0.27
min	0.0	0.0
25%	0.4	0.28
50%	0.65	0.51
75%	0.86	0.75
max	1.0	1.0
unique	361 (1.8%)	363 (1.82%)
most common	0.89722 (2.51%)	0.23204 (1.47%)
missing	0 (0.0%)	0 (0.0%)
infinite	0 (0.0%)	0 (0.0%)



33bis

Axes d'amélioration

- **Créer une API + Interface**
  (pour segmenter en temps réel les nouvelles données en attendant le nouveau modèle périodique)
- **Utiliser des tests statistiques**
  (Hopkins statistic, Distance distribution)
- **Tester l'influence de la quantité de données sur le silhouette score**
  (pour trouver la période idéale pour le modèle initial puis ensuite lors des premiers re-entrainements et enfin pour savoir si l'on doit garder les anciennes données)

Merci de m'avoir écouté, évalué et conseillé.

olist store