Olist

Segmentez les clients d’un site e-commerce

Contexte

Enjeux

L'equipe Marketing Olist cherche à :

- **Fidéliser** les clients existants
- **Aumenter** leur panier moyen

Dans tous les cas, l'équipe doit communiquer de maniére:

- **Pertinente**: communication doit être adaptée à la typologie de client
- **Automatisée**: afin d'optimiser le travail de personnalisation, il faut être capable de cibler des groupes cohérents de clients

Objectifs

Nous allons ici chercher à créer un **modèle de classification non supervisé** permettant de répondre aux questions :

- **Combien** y'a-t-il de segments de clients?
- Qu'est-ce qui **caractérise** ces segments?
- A quelle fréquence fqut-il les mettre à jour?

Méthode

Nous utilisons le jeu de données issu de la compétition Kaggle Brazilian E-Commerce Public Dataset by Olist, portant sur l'historique de **99441 commandes** effectuées par **96096 clients** sur la platforme Olist entre 2016 et 2018.
Nous sommes donc face à un **problème de classification non-supervisé**.

Feature engineering

- Nous allons créer les trois variables **RFM (Recency, Frequency et Monetary)** qui permmetent d'effectuer une description pertinente des clients:
    - *Recency:* nombre de **jours depuis le dernier achat du client**
    - *Frequence*: **nombre total de commandes** effectuée par le client
    - *Monetary*: montant moyen des commandes effetuées par le client
- Nous allons enricher notre analisis en creant des nouvelles variables pertinentes

- Nous allons **préparer les données** afin **d'entraîner plusieurs modèles**, puis les comparer.

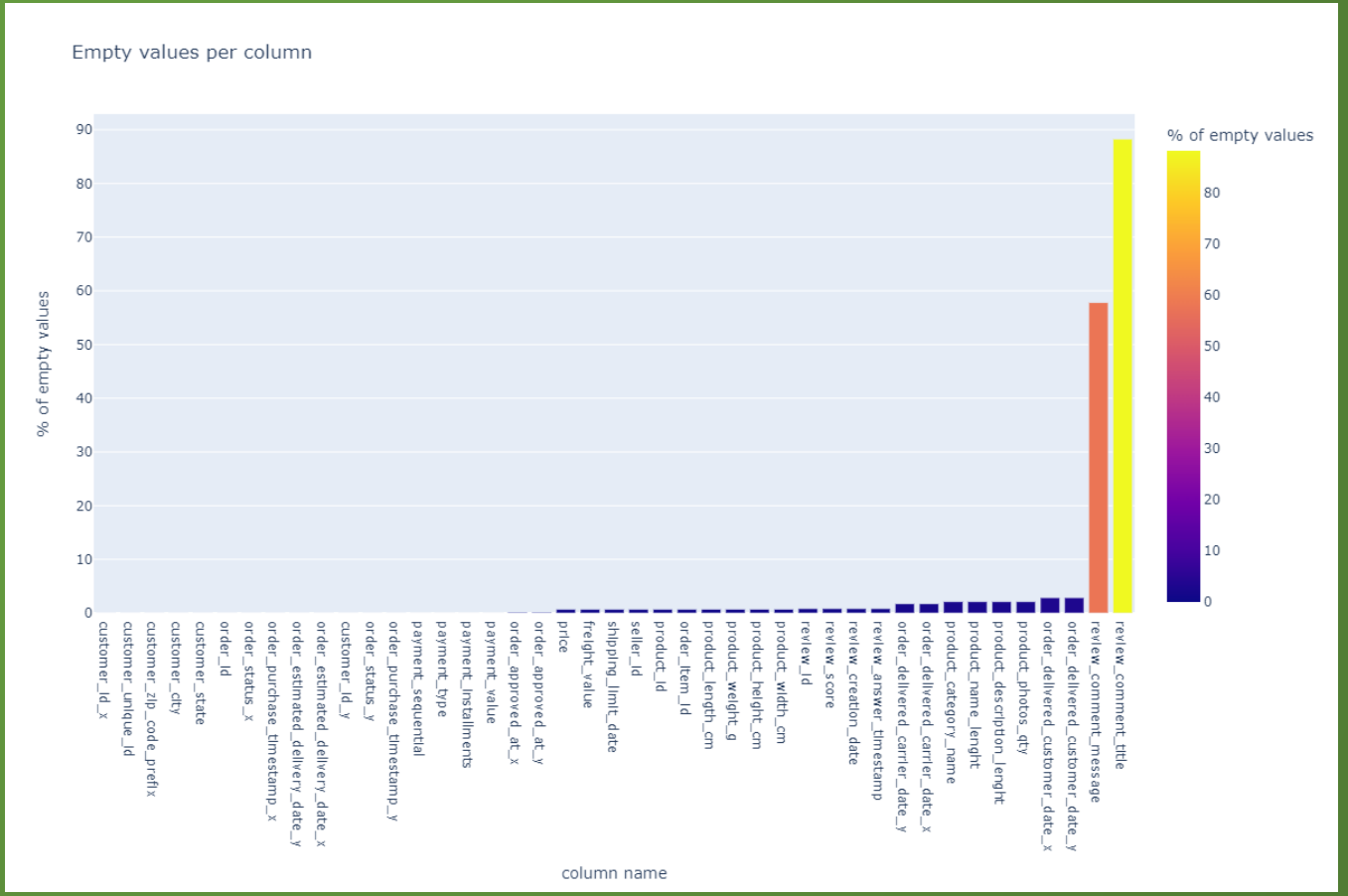
- Enfin, nous allons tester **notre meilleur modèle**:
    - **expliquer sa segmentation** prédiction en visualisant les differences de distribution entre segments
    - Mesurer l'évolution de sa performance dans le temps et proposer un devis de contrat de maintenance.

Qualité générale des données

Le jeu de données est composé 9 fichiers avec, en total, **44 variables.**

- Nous utilisons la fonction de **merge** pour compiler tous les dataframes en utilisant la feature commune

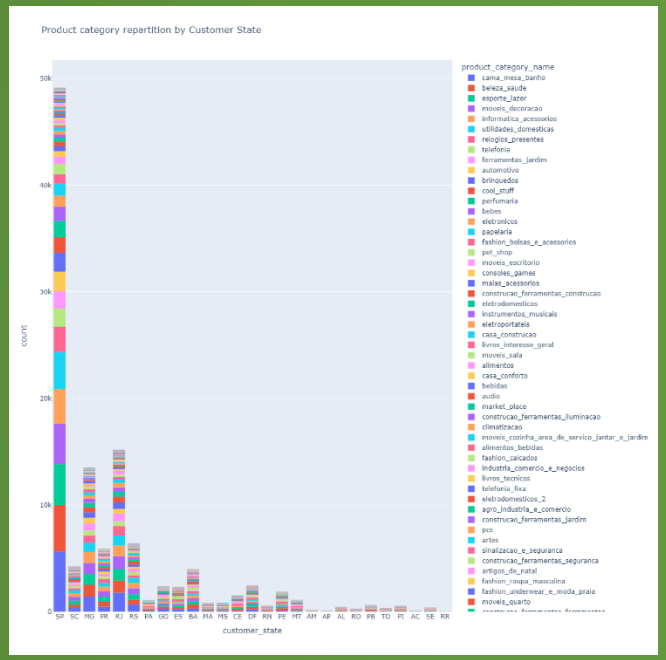
- La plus part des variables ont moins de 3% de valeurs vide.
- La variable review_comment title et review_comment_message présente des valeurs manquant tres haut que nous supprimons.



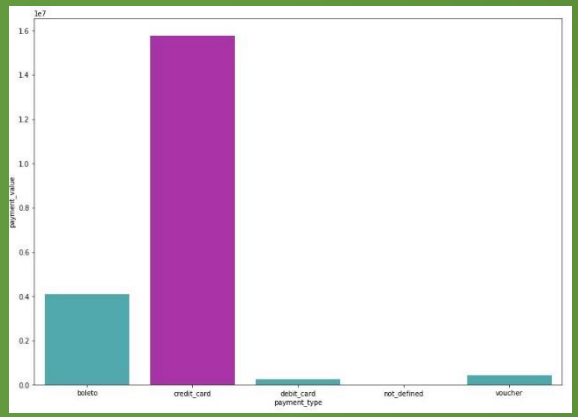
Analyse exploratoire de donnes (EDA)

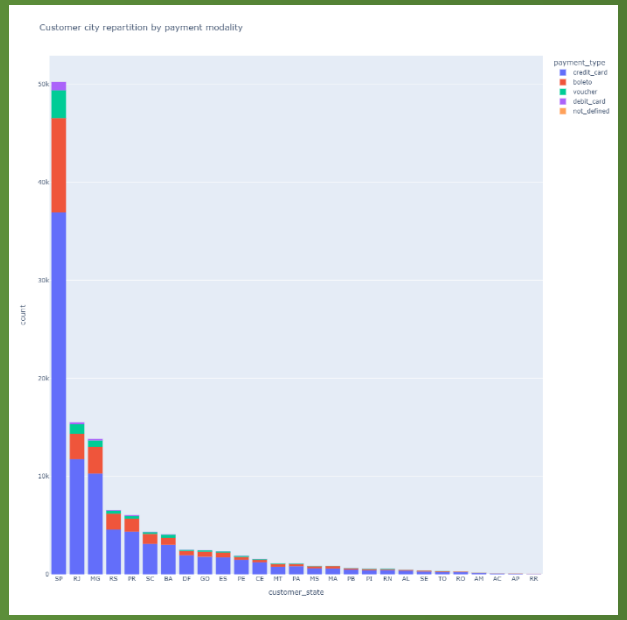
Nous pouvons voir les produits les plus vendus par l'entreprise

Les États qui utilisent le plus cette entreprise pour acheter différents produits.

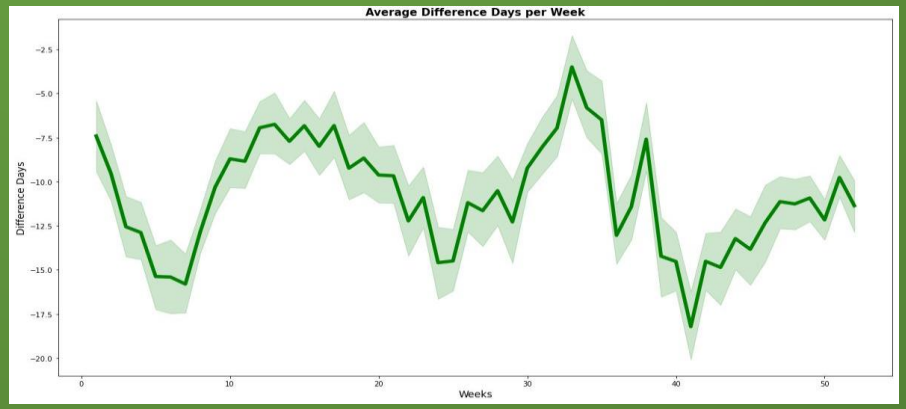


Le mode de paiement préféré en termes demontant total de l'argent





le délai de livraison et l'estimation du nombre moyen de jours de différence par semaine



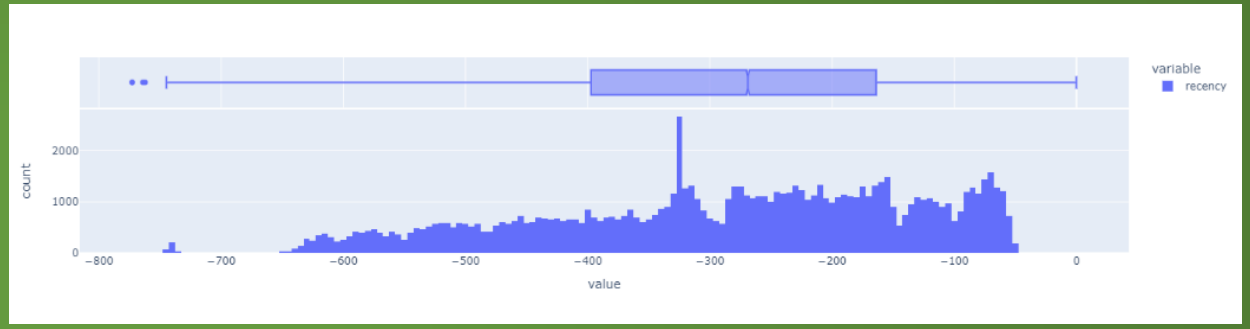
Vartiable RFM et transformation

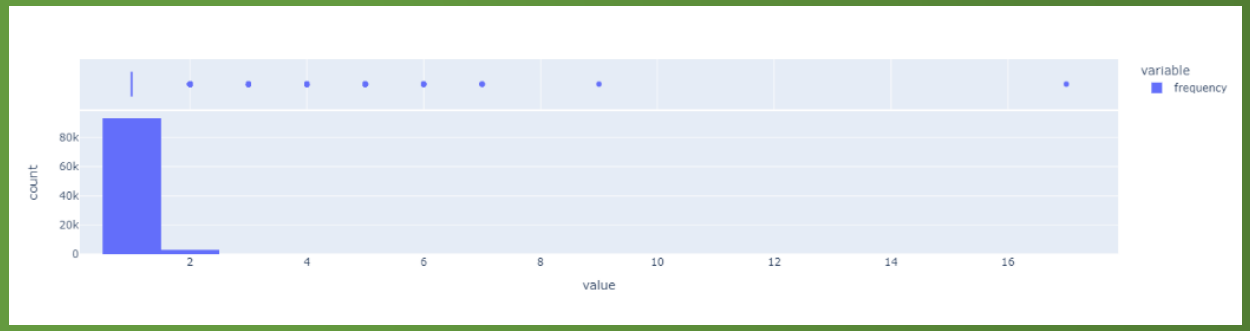
- Parmis toutes les variables disponibles, nous ne regarderons que les **donnees RFM et variables conaisance matier.**

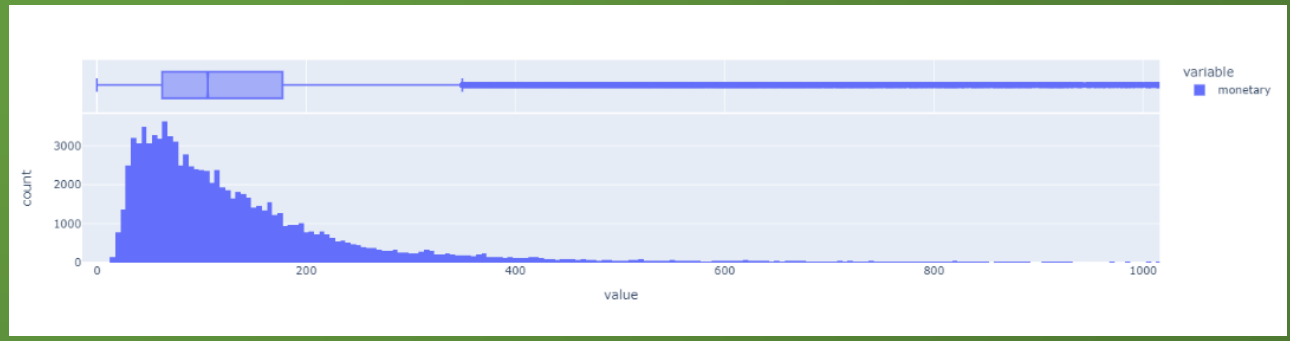
- Il n'y a pas de doublons, ni valeurs vides, ni valeur impossibles (sauf quelques outliers que nous eliminons).

- Les variables son toutes **desequilibrees**

- Nous appliqouns un **log** a chaque variable pour reduir les distances entre donnees tres eloignees de 0

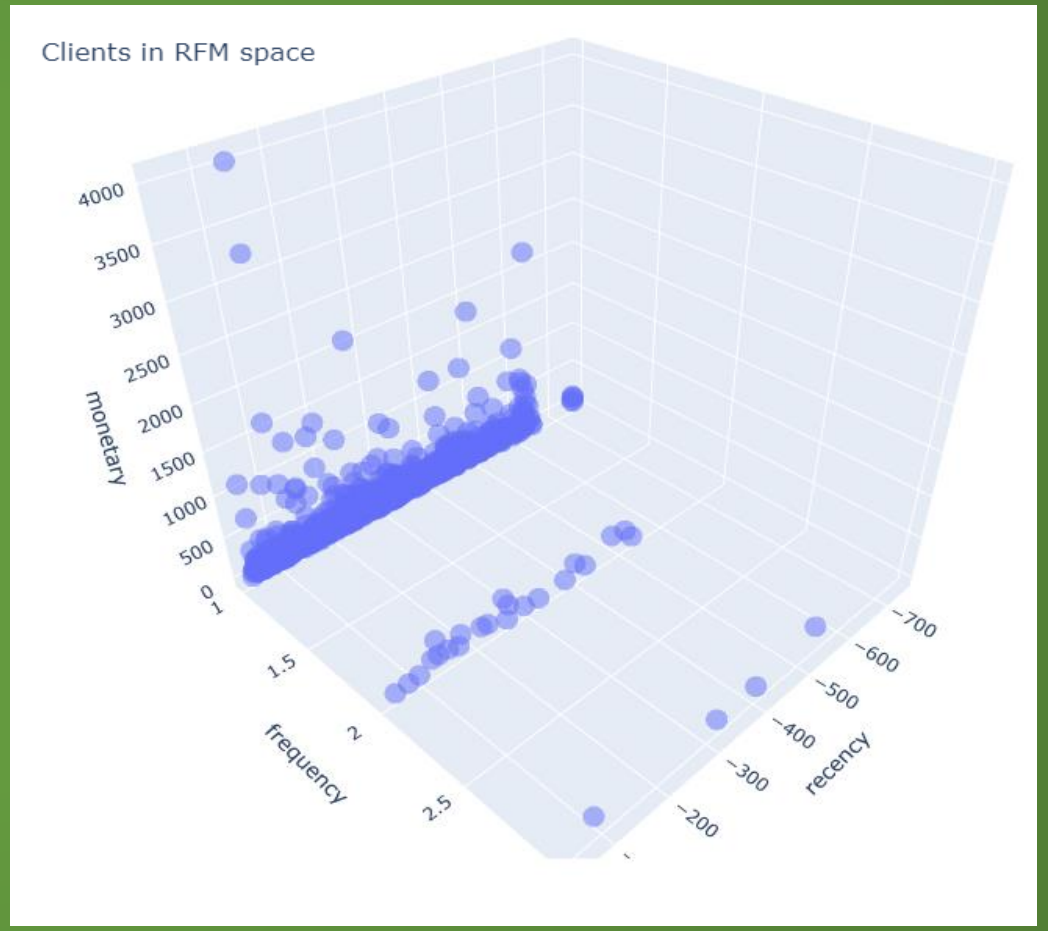






Visualisation des donnees

A niveaux de variable RFM, la **frequence separe bien** les clients



Modeles de classification non-supervisé

Nous utilisons 3 modèles de clustering non supervisés pour les variables RFM :

- **K-means:** est un algorithme de clustering basé sur les centroïdes ou sur les partitions. Cet algorithme partitionne tous les points de l'espace échantillon en K groupes de similarité (mesurée à l'aide de la distance euclidienne).

- **Agglometative clustering:** est une famille générale d'algorithmes de clustering qui construisent des clusters imbriqués en fusionnant successivement des points de données.

- **DBSCAN:** est un algorithme de clustering basé sur la densité. Le fait essentiel de cet algorithme est que le voisinage de chaque point d'un cluster qui se trouve dans un rayon donné (R) doit avoir un nombre minimum de points (M). Cet algorithme s'est avéré extrêmement efficace pour détecter les valeurs aberrantes et traiter le bruit.

Evaluation des modeles de classification non-supervisé

Nous avons essayé de segmenter nos clients à l'aide de différents modèles, et pour chacun de ces modèles nous avons essayé de trouver les hyper-paramètres qui donnaient les meilleurs résultats.

Nous recherchons un modèle stable, rapide à entraîner et à évaluer, et simple à interpréter : il doit comporter peu de clusters (moins de 10), suffisamment équilibrés et facilement différenciables.

- **model**: Nom du modèle
- **n_clusters:** Nombre de clusters trouvés
- **labels**: Liste des étiquettes des clusters prédits
- **cluster_centers**: Liste des coordonnées des centres de clusters
- **Inertia**: Liste des valeurs d'inertie des clusters
- **Time**: Temps passé pour l'entraînement et la prédiction

METRICS

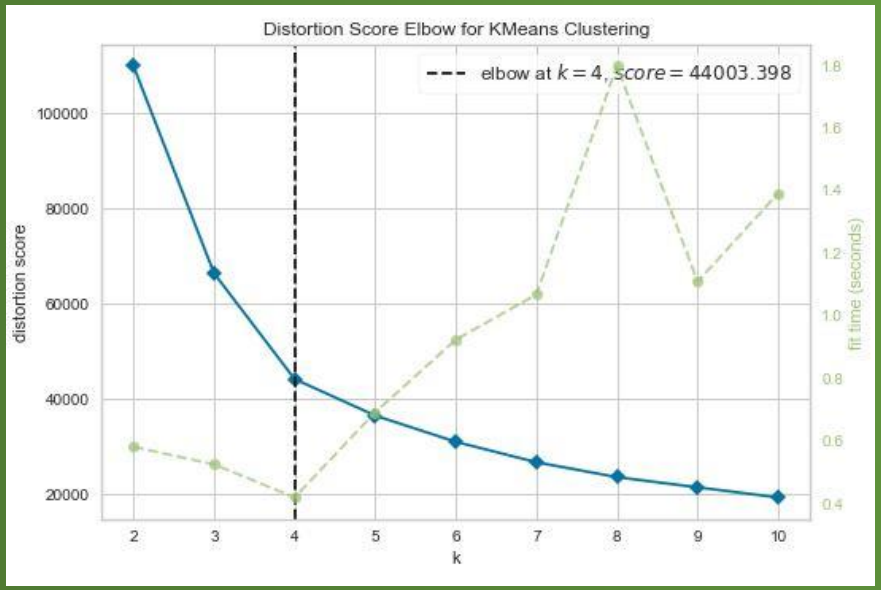
- **silhouette_score:** calculée en utilisant la distance moyenne intra-groupe (a) et la distance moyenne la plus proche (b) pour chaque échantillon.
- **davies_bouldin_score:** Le score est défini comme la mesure de similarité moyenne de chaque cluster avec son cluster le plus similaire, où la similarité est le rapport entre les distances intra-cluster et les distances inter-cluster. Ainsi, les clusters les plus éloignés et les moins dispersés obtiendront un meilleur score.
- **calinski_harabasz_score:** The score is defined as ratio between the within-cluster dispersion and the between-cluster dispersion.
- **meta_score:** somme des scores standardisés

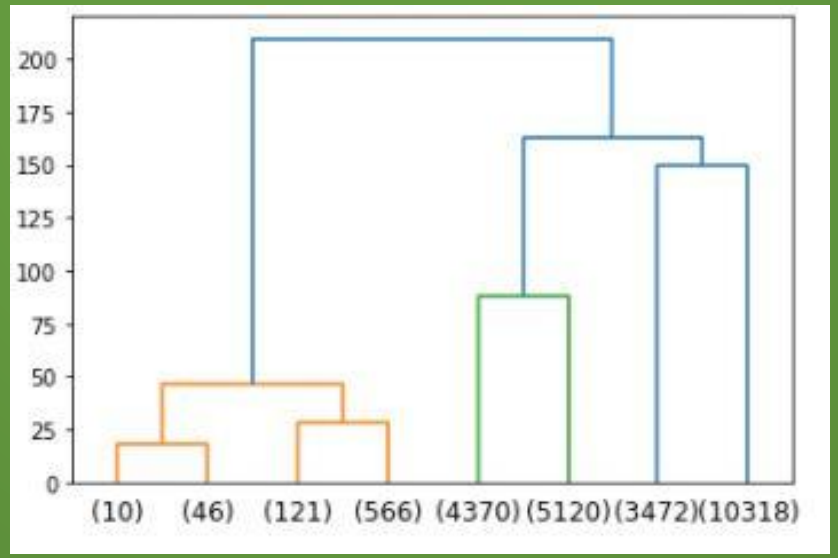
Estimation du nombre de segments

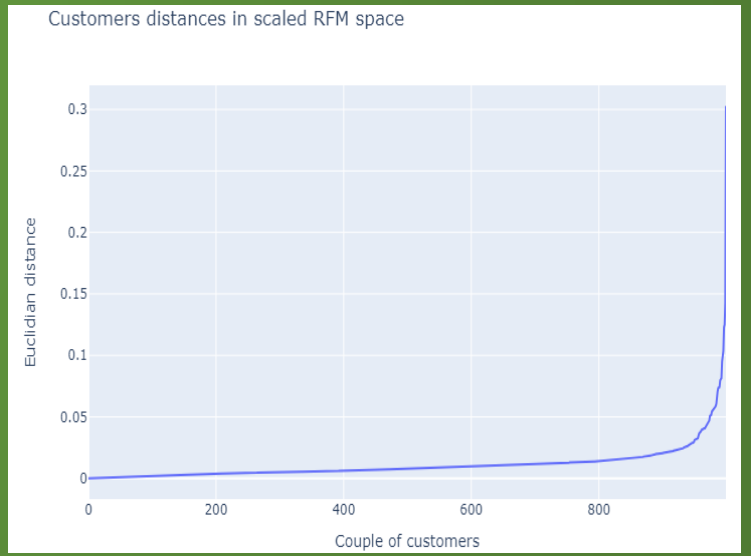
La méthode du coude utilisant Kmeans nous montre que 4 clusters sont suffisants pour diviser les données en groupes appropriés.

Dendogramme (methode de Ward) nous indique que le meilleure segmentation se fera avec 4 clusters (bleu)

En utilisant la méthode du coude, nous voyons que la distance euclidienne entre deux points est inférieure à 0.025 pour 95% des couples de points.



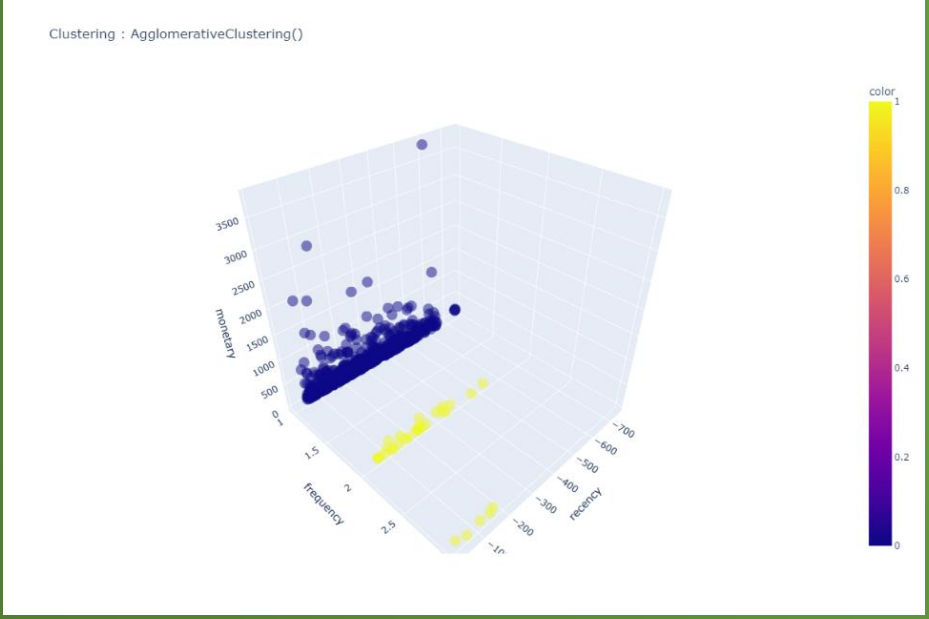




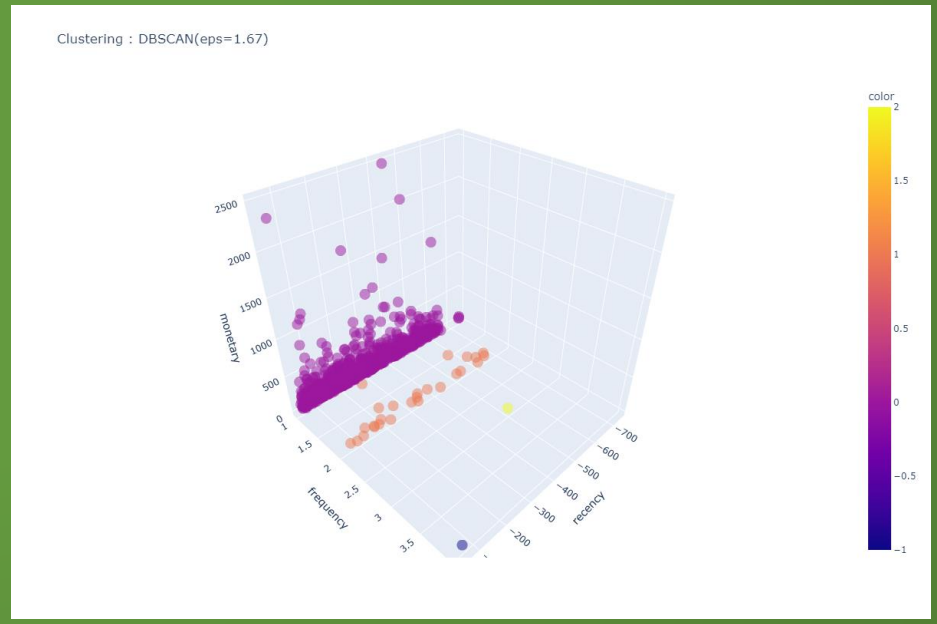
Modeles de classification non-supervisé

```
models_results_rfm.sort_values(by='silhouette_score', ascending=False)
```

	model	n_clusters	labels	cluster_centers	inertia	time	silhouette_score	davies_bouldin_score	calinski_harabasz_score	meta_score	standard_calinski_harabasz_score	standard_davies_bouldin_score	standard_silhouette_score
1	AgglomerativeClustering()	2	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	None	None	4.173812	0.702671	0.491942	5549.141147	3.155757	-2.233491	-2.578964	2.810285
2	DBSCAN(eps=1.67)	3	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	None	None	1.363387	0.682200	0.805012	2009.015058	2.093405	0.412756	-0.948733	0.731916
0	KMeans(n_clusters=4)	4	[0, 0, 3, 3, 1, 3, 2, 2, 0, 0, 0, 1, 0, 0, 0, ...	[[0.5715794813590631, -0.17359306829690505, -0...	44003.594416	0.000009	0.377132	0.777229	36140.641602	3.440281	0.618809	-0.633594	2.187877

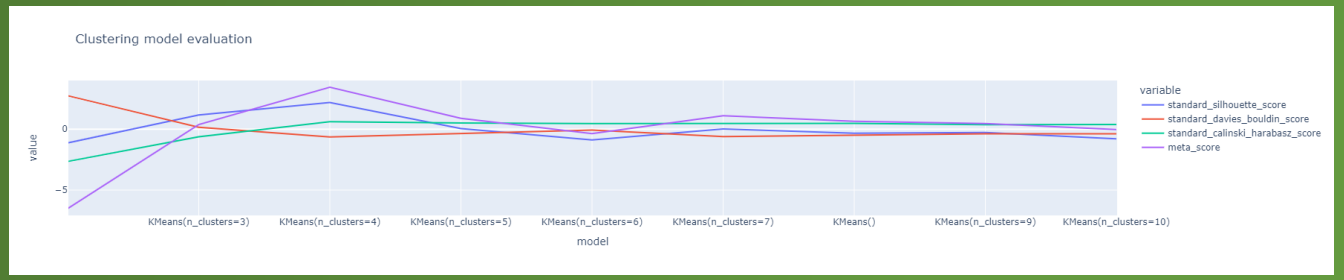


- Cluster 0 : les clients qui ont effectué 1 achat.
- Cluster 1 : clients ayant effectué plus d'un achat (achat supérieur à la moyenne du panier).

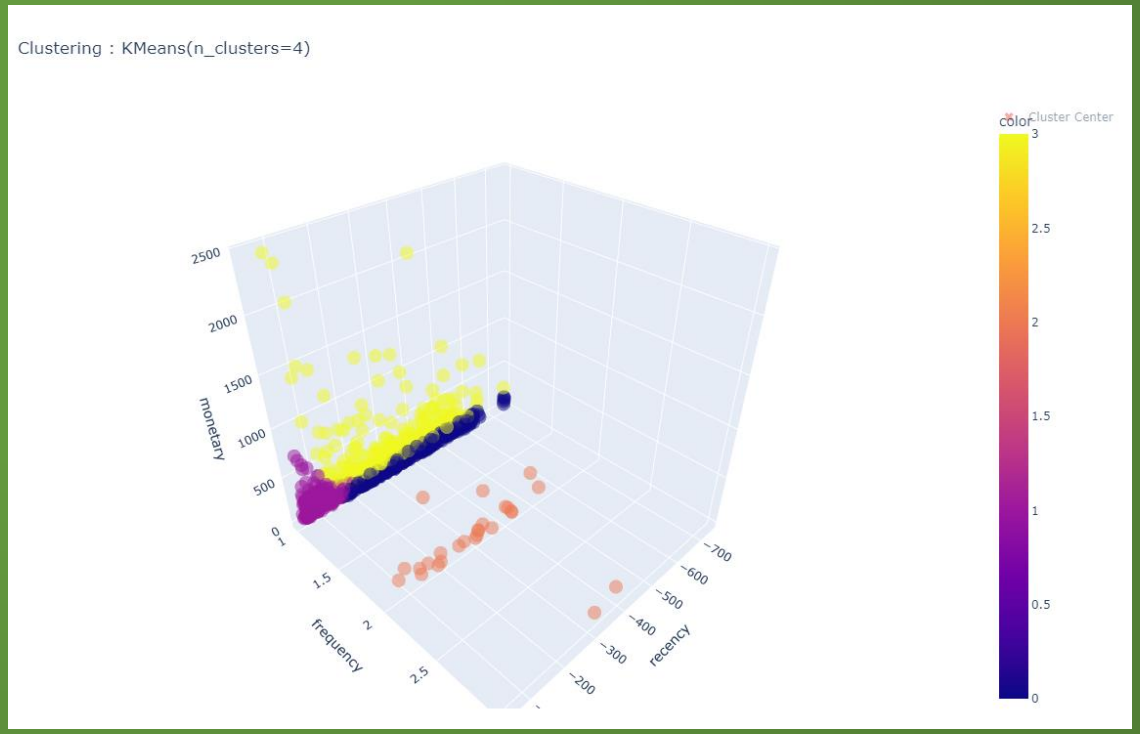


- Cluster -1 : un artefact du modèle : un seul individu isolé.
- Cluster 0 : clients ayant effectué 1 achat.
- Cluster 1 : clients ayant effectué 2 achats.
- Cluster 2 : clients ayant effectué plus de 2

Modele de classification retenu: Kmeans(4)

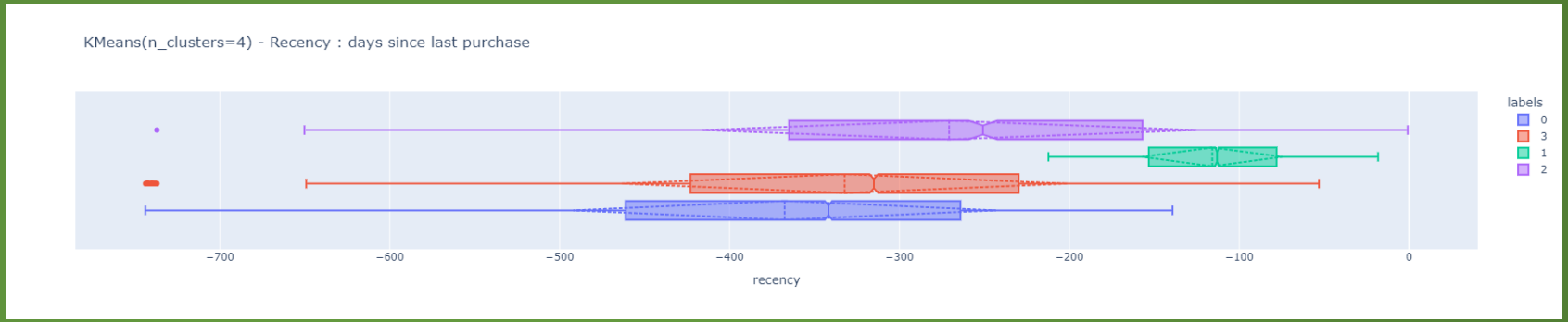


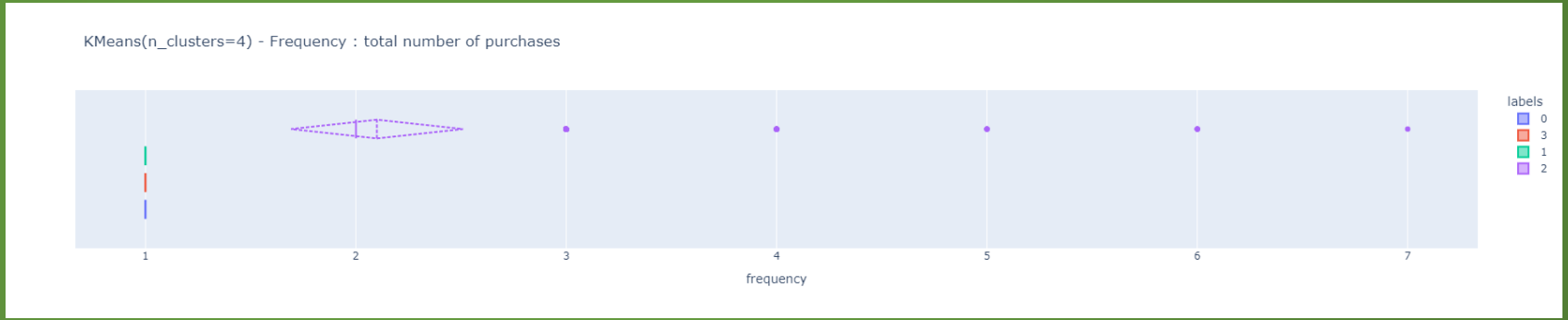
Ce modele est **pertinent**, **rapide** a entrainer

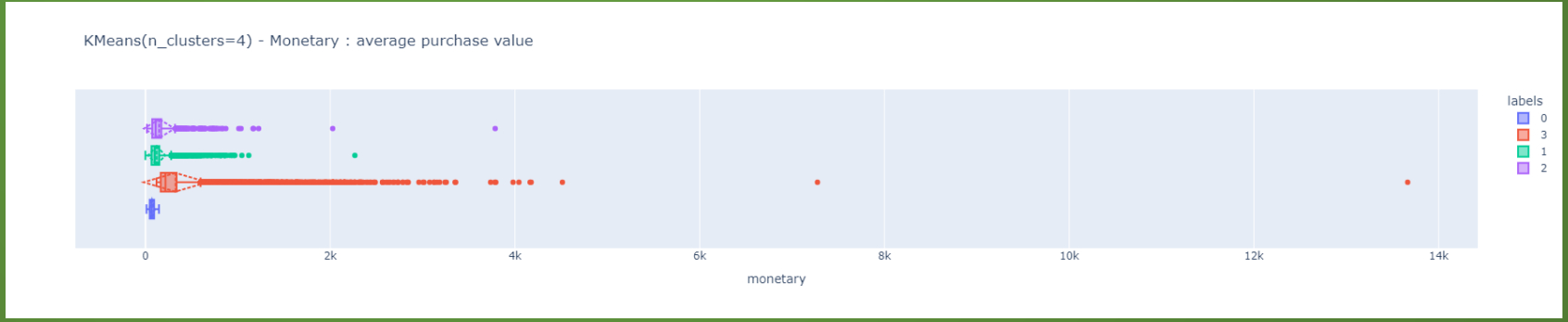


Segments de clients identifies

- **cluster 0** : les anciens clients avec un panier moyen faible.
- **cluster 1** : les clients les plus récents.
- **cluster 2** : les clients qui ont fait plus d'un achat.
- **cluster 3** : anciens clients avec un panier moyen élevé.







Proposition de contrat de maintenance

Nous allons observer la performance de notre modèle au cours du temps en fonction de sa fréquence de mise à jour en utilizant la funtion adjusted_rand_score de sklearn metrics.

ARI = (RI - Expected_RI) / (max(RI) - Expected_RI).