Projet 4

Segmentez des clients d’un site e-commerce

par Rouba Yaacoub

Quelle e-commerce nous travaillons avec ??

Cette fois çi, nous travaillons avec olist, une startup brésilienne qui opère dans le secteur du commerce éléctronique, principalement dans le marketplace.

Notre mission

Suite aux informations fournies par olist sur l'historique de commandes, les produits achetés, les commentaires de satisfaction, et la localisation des clients depuis janvier 2017, nous allons:

1. comprendre **les différents types d'utilisateurs**
2. regrouper **l'ensemble des clients de profils similaires**
3. proposer **un contrat de maintenance**.

TABLE OF CONTENTS

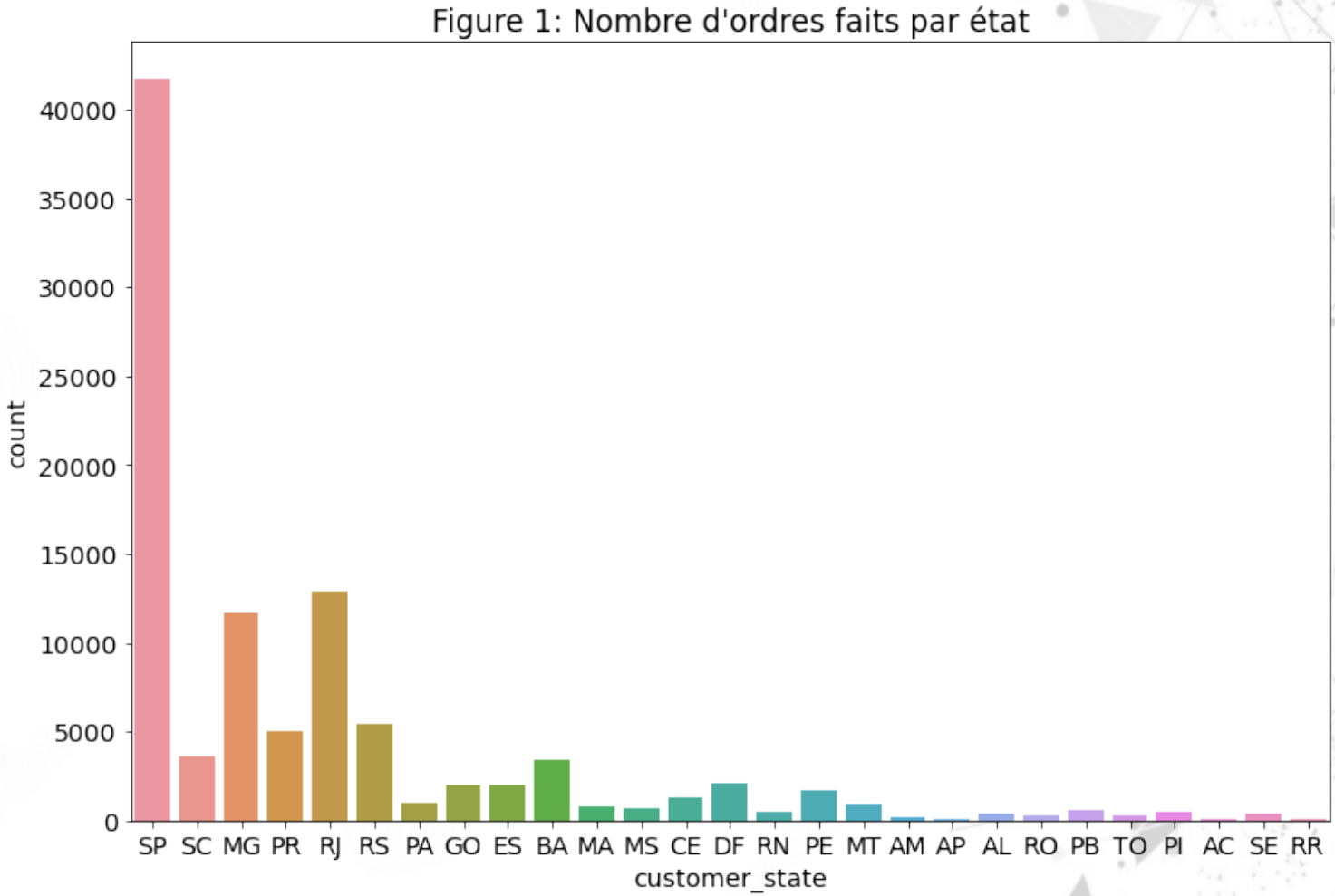
olist

Plan

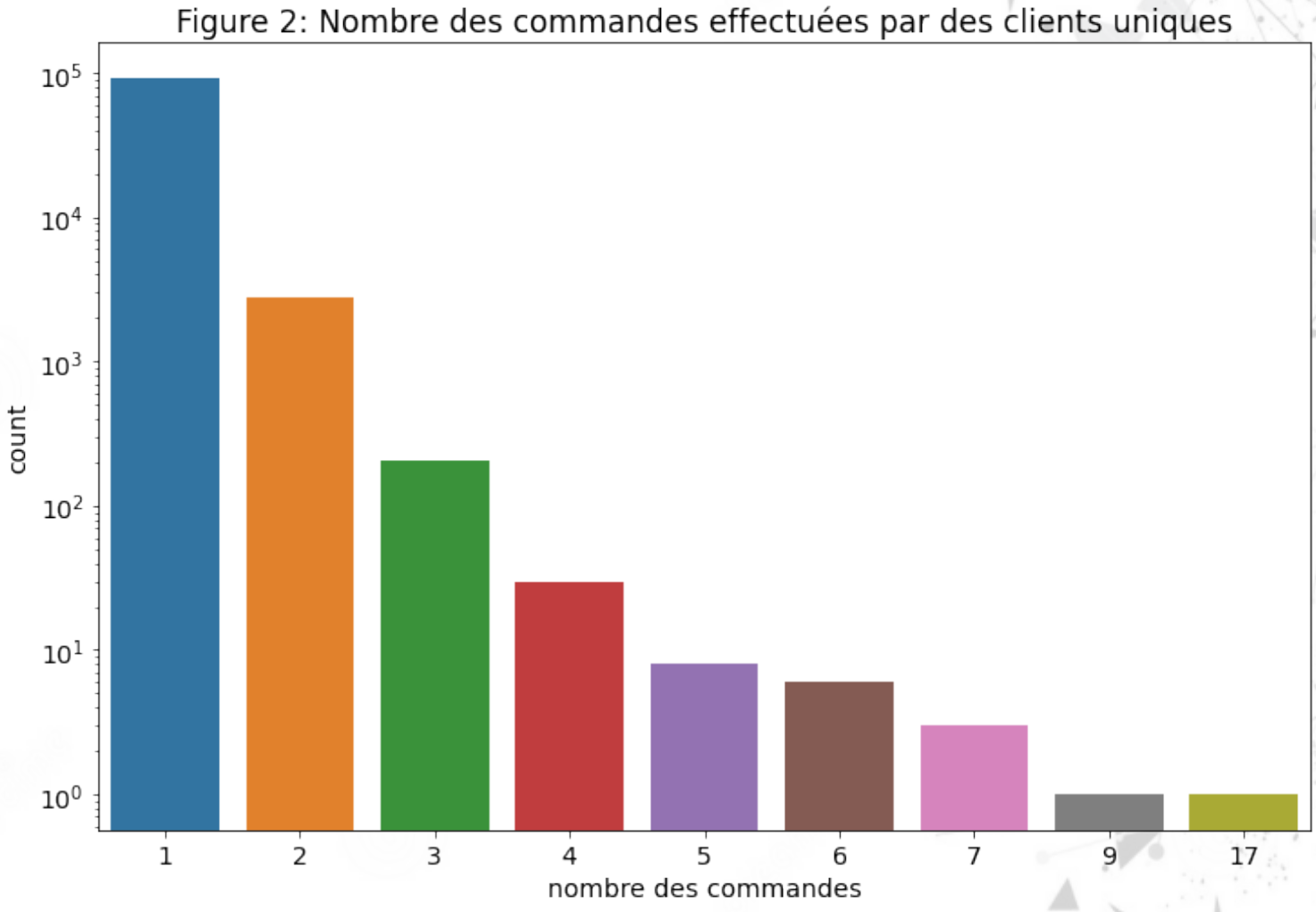
Lire les fichiers csv et les nettoyer des valeurs aberrantes:

Des analyses profondes ont été faites afin des nettoyer les dataframes, les données sont propres et prêtes à être explorées.

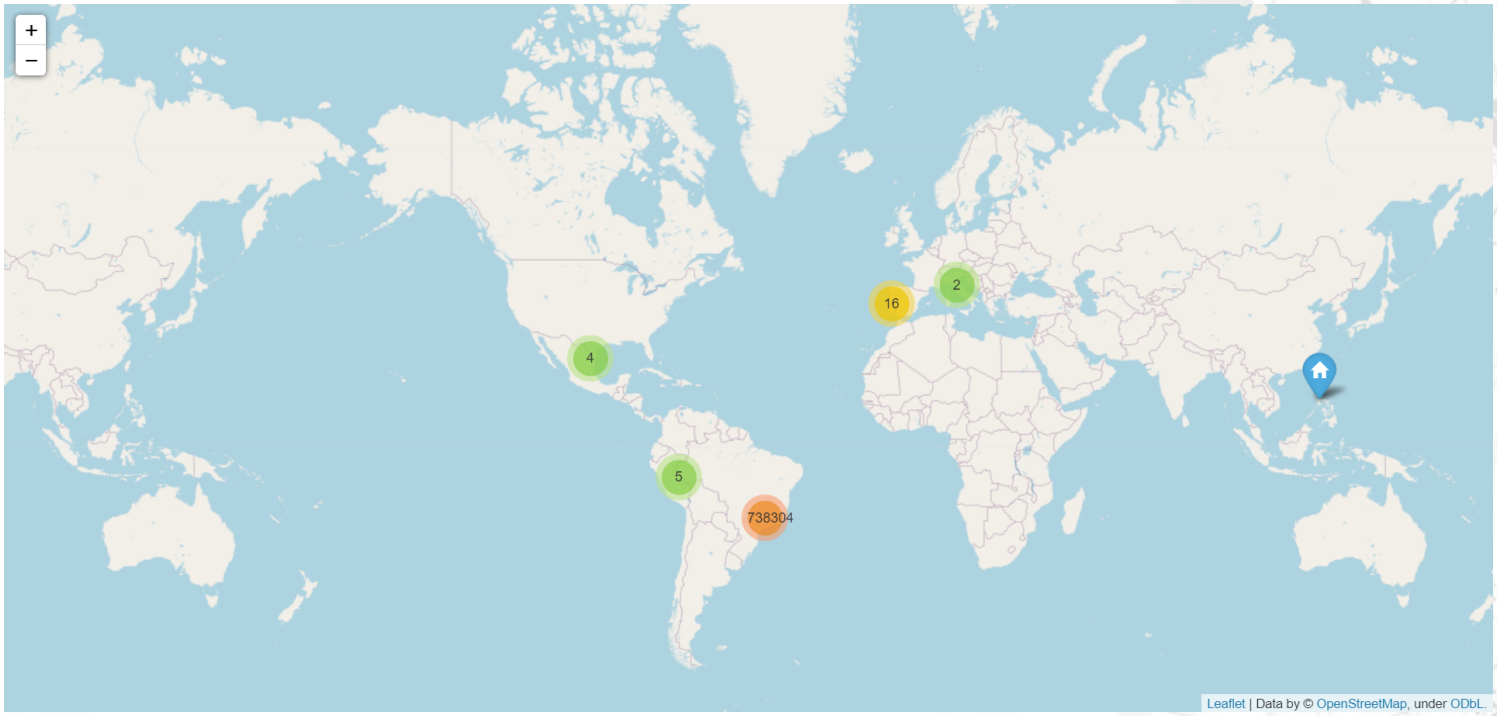
Dataframe "olist customers dataset.csv":



Dataframe "olist customers dataset.csv":



Dataframe "olist geolocation dataset.csv":



Dataframe "olist geolocation dataset.csv":

Dataframe "olist order items dataset.csv":

Figure 3: Le pourcentage des articles achetés par an

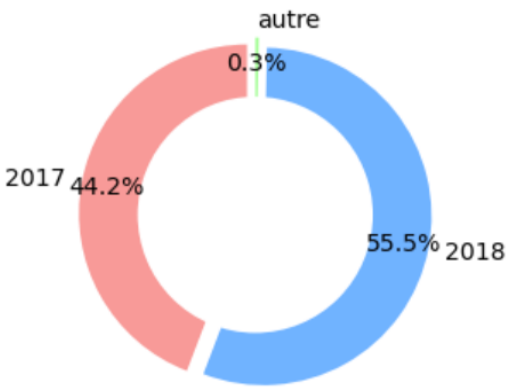
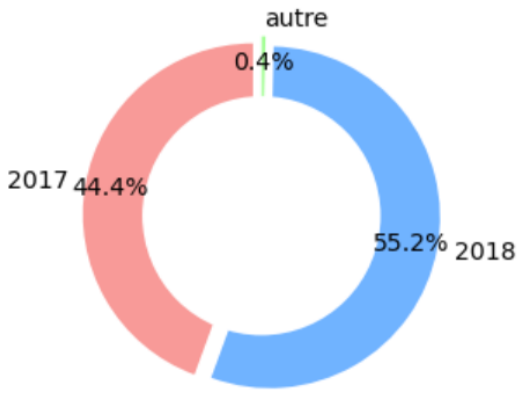
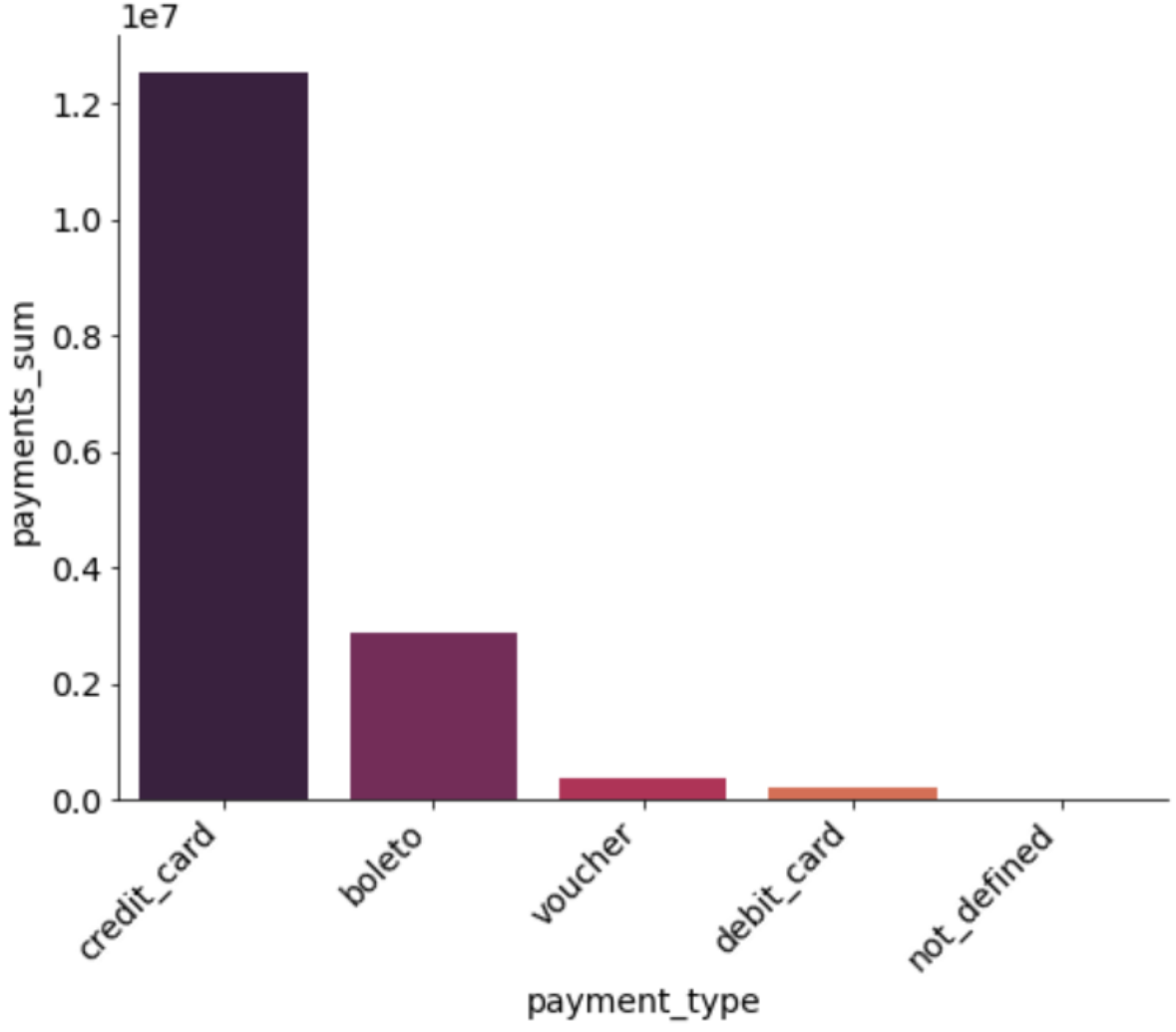


Figure 4: Le pourcentage du prix des produits acheté par an



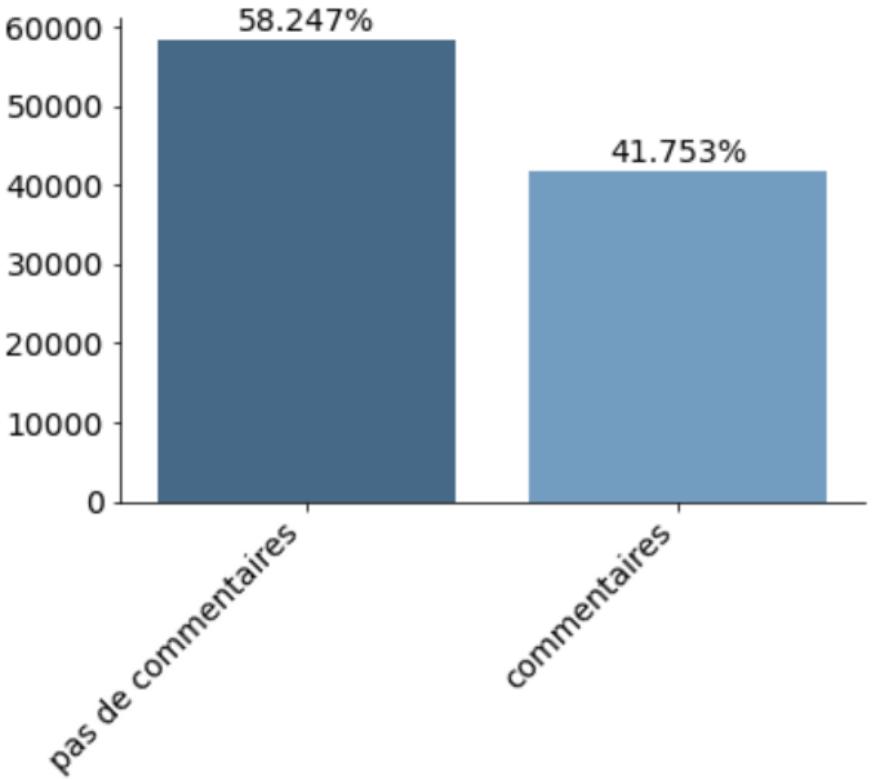
Dataframe "olist order payments dataset.csv":

Figure 5: La chart de nombre des achats par type de paiement



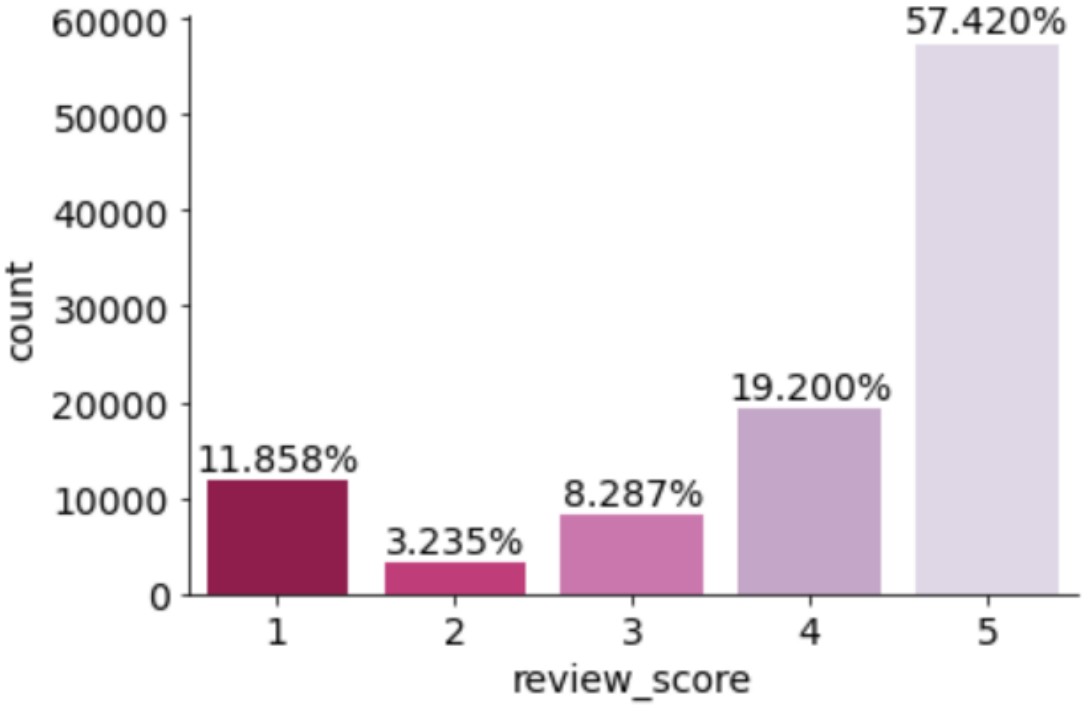
Dataframe "olist order reviews dataset.csv":

Figure 6: Le nombre de commandes qui ont reçu une note de critique ou non



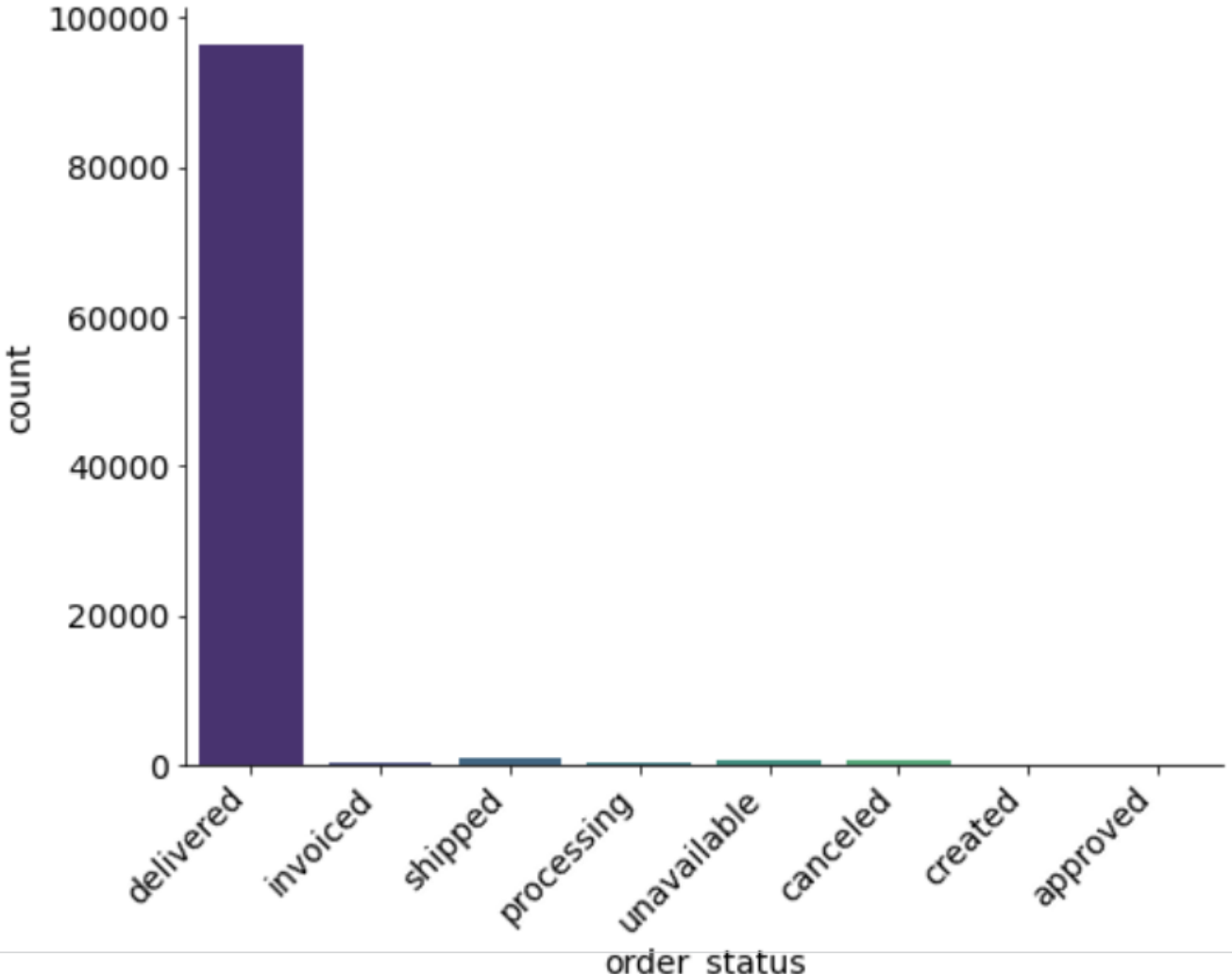
Dataframe "olist order reviews dataset.csv":

Figure 7: Le nombre de commandes vs. une note de critique

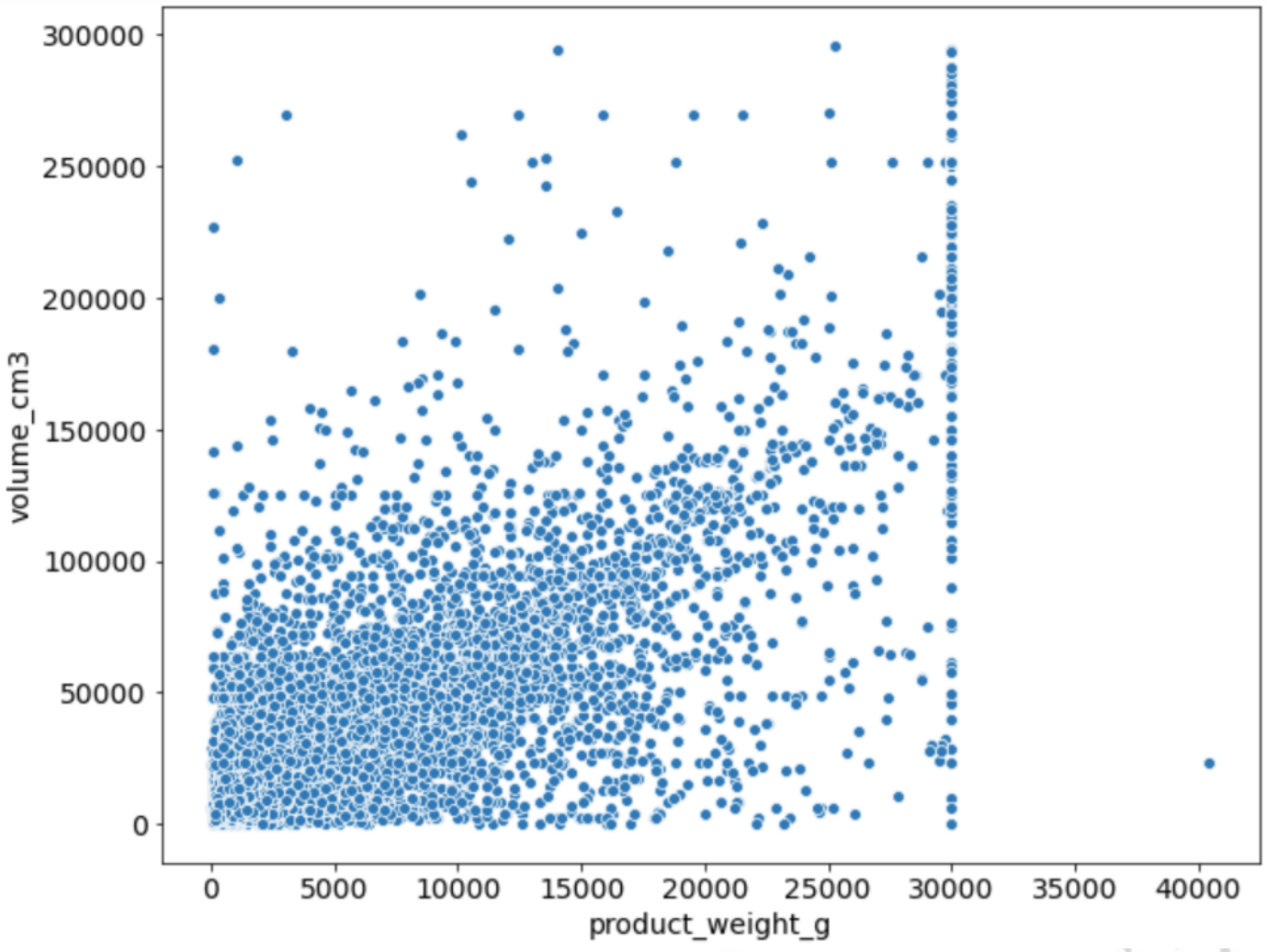


Dataframe "olist order payments dataset.csv":

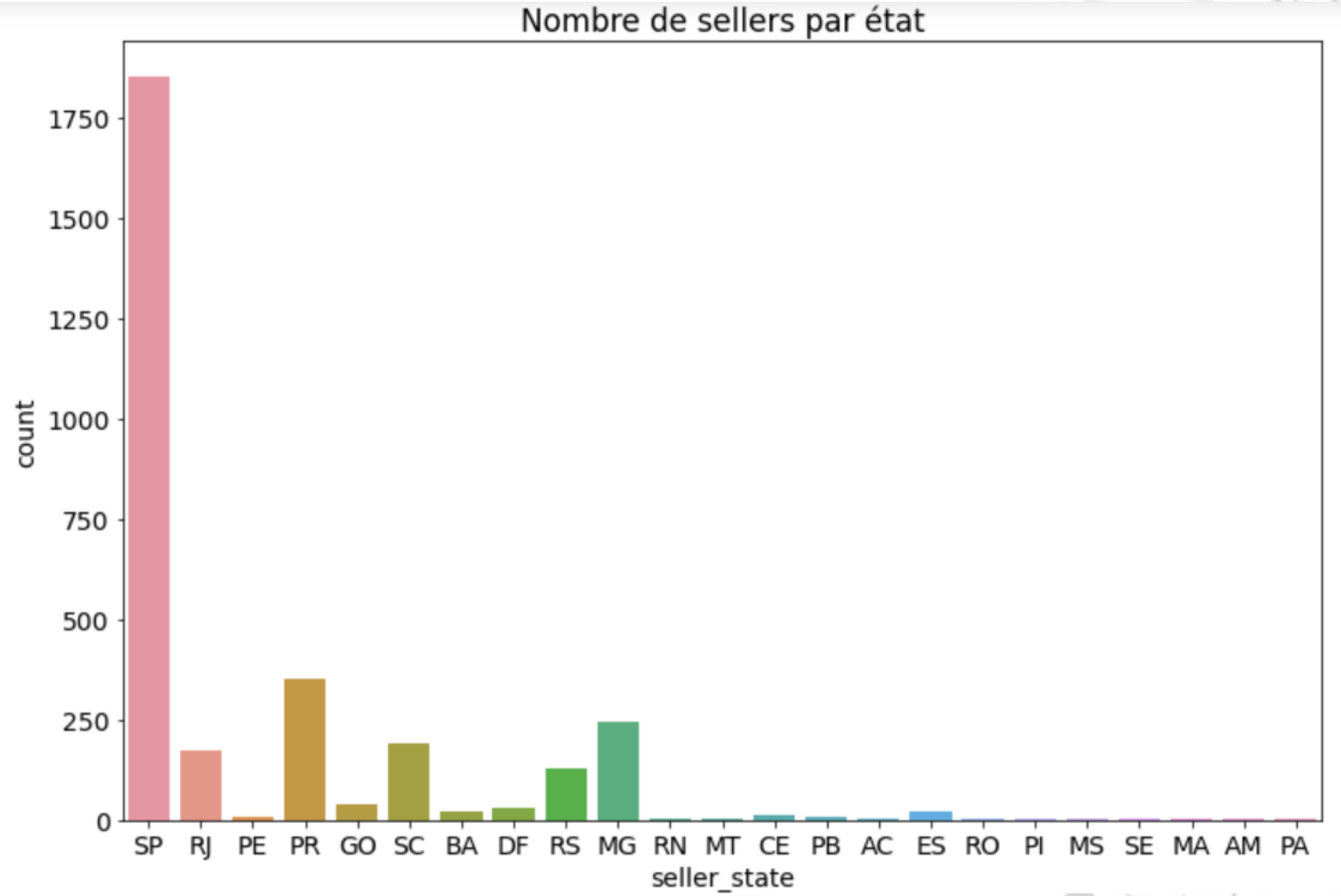
Figure 8: Le nombre de commandes vs le status de la commande



Dataframe "olist products dataset.csv":



Dataframe "olist products dataset.csv":

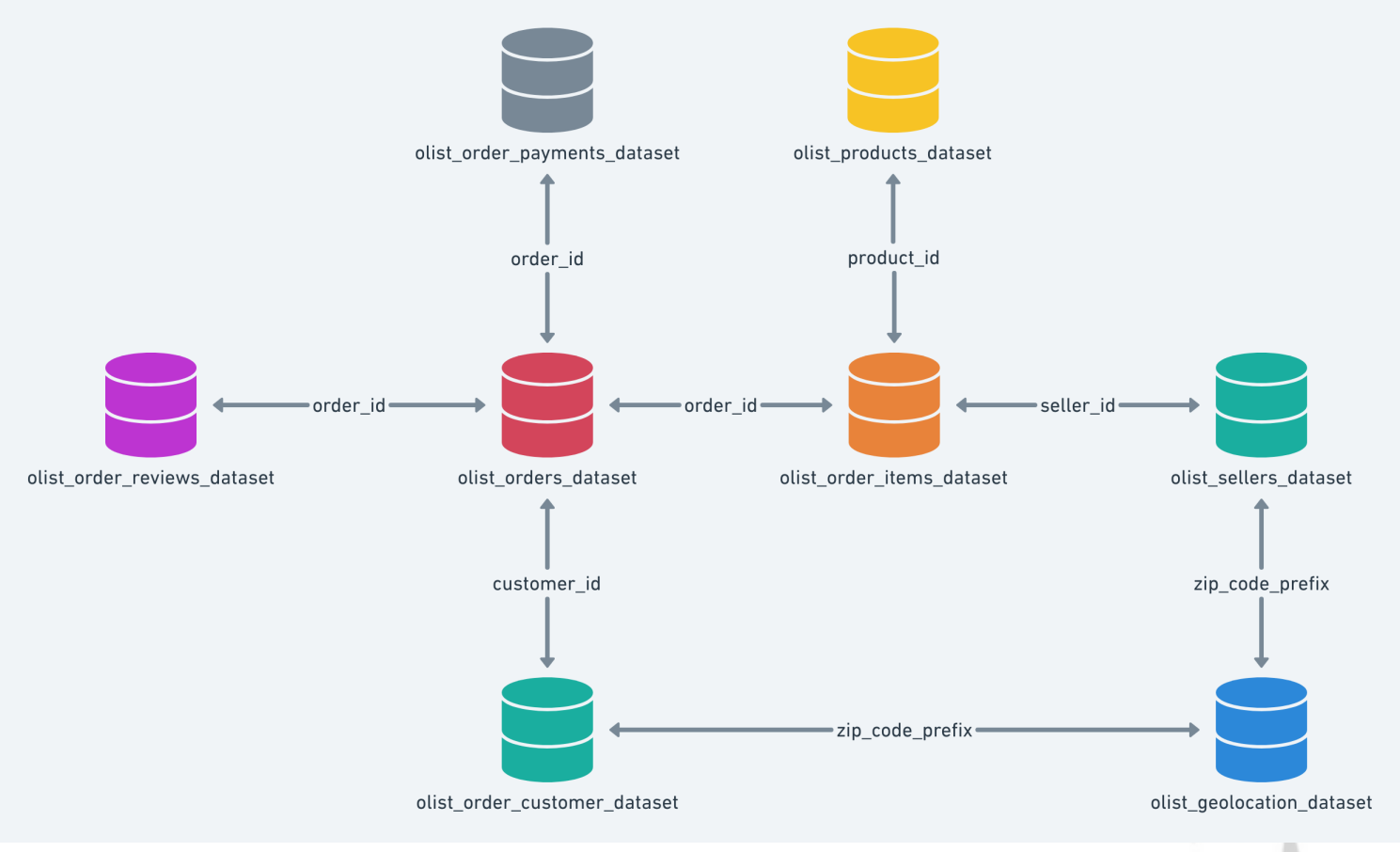


Plan

Fusion de data

Comme nous pouvons le voir les data sont divisées en plusieurs sous-data pour une meilleure compréhension et organisation.

Le schéma ci-dessous explique comment les joindre :



Plan

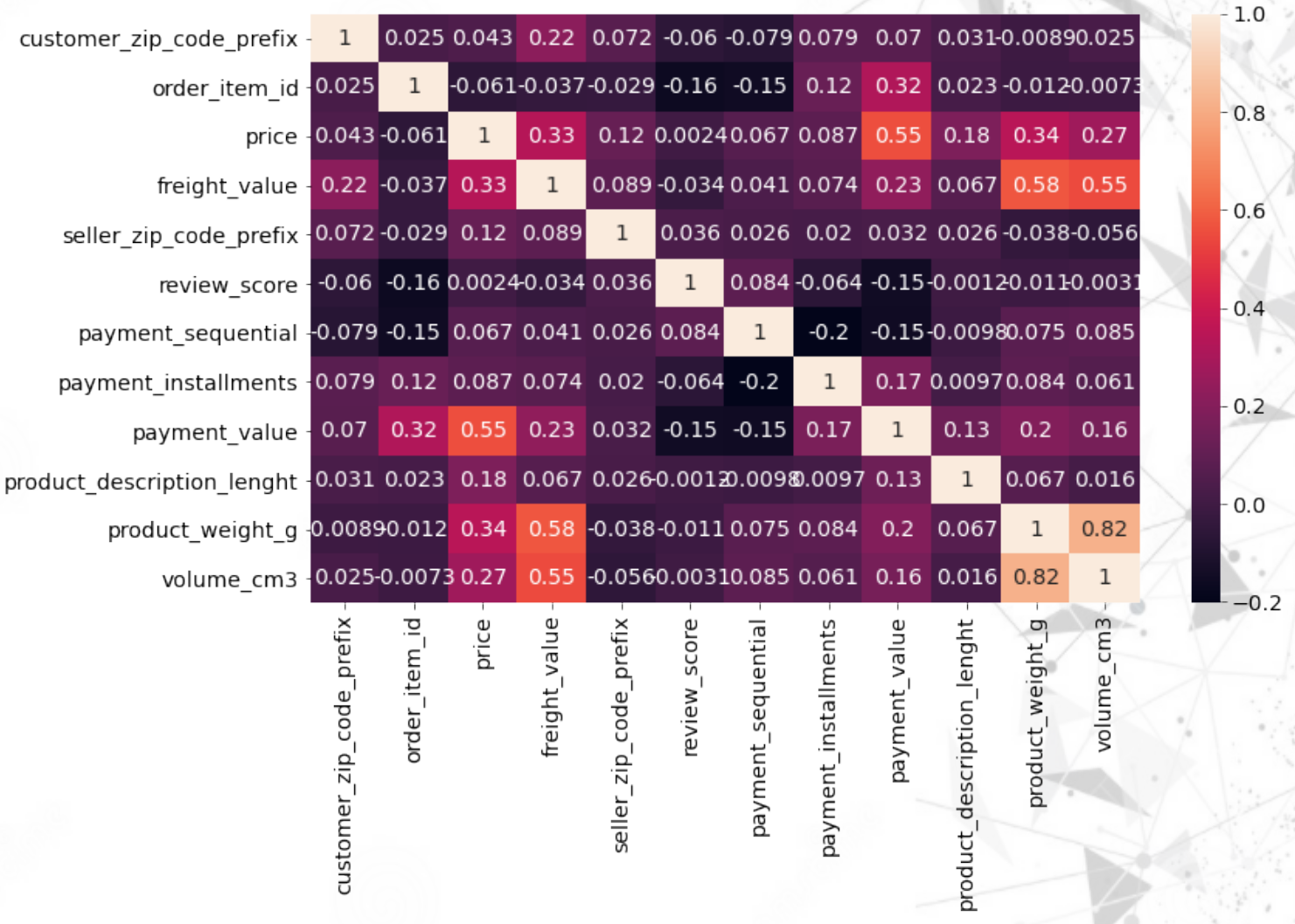
Variables pertinentes – corrMatrix

olist

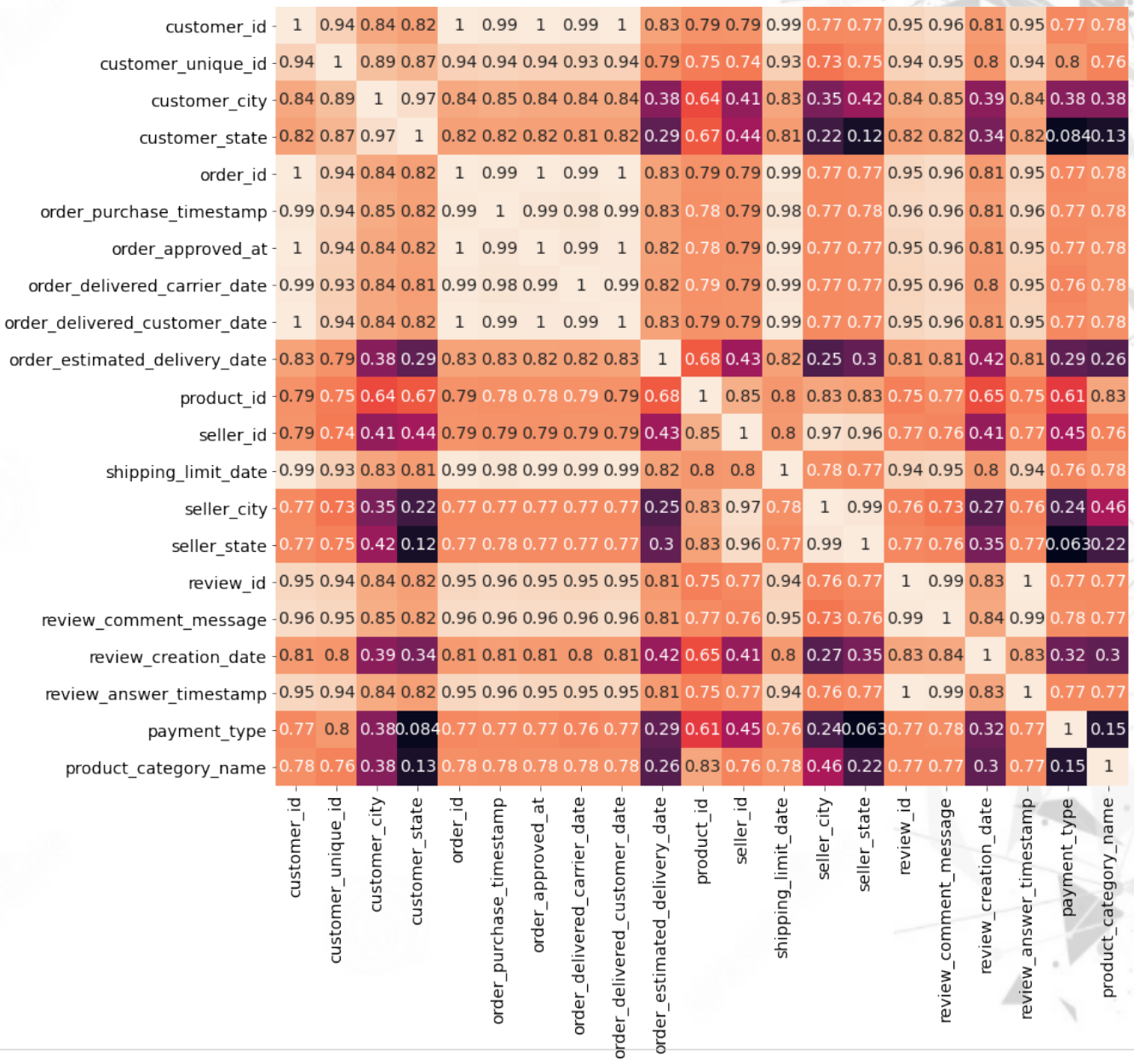
```
In [37]: data_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 115728 entries, 0 to 115727
Data columns (total 37 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   customer_id                    115728 non-null  object
 1   customer_unique_id             115728 non-null  object
 2   customer_zip_code_prefix       115728 non-null  int64
 3   customer_city                  115728 non-null  object
 4   customer_state                 115728 non-null  object
 5   order_id                       115728 non-null  object
 6   order_status                   115728 non-null  object
 7   order_purchase_timestamp       115728 non-null  object
 8   order_approved_at              115713 non-null  object
 9   order_delivered_carrier_date   115726 non-null  object
 10  order_delivered_customer_date  115720 non-null  object
 11  order_estimated_delivery_date  115728 non-null  object
 12  order_item_id                  115728 non-null  int64
 13  product_id                     115728 non-null  object
 14  seller_id                      115728 non-null  object
 15  shipping_limit_date            115728 non-null  object
 16  price                          115728 non-null  float64
 17  freight_value                  115728 non-null  float64
 18  seller_zip_code_prefix         115728 non-null  int64
 19  seller_city                    115728 non-null  object
 20  seller_state                   115728 non-null  object
 21  review_id                      115728 non-null  object
 22  review_score                   115728 non-null  int64
 23  review_comment_title           13751 non-null   object
 24  review_comment_message         48961 non-null   object
 25  review_creation_date           115728 non-null  object
 26  review_answer_timestamp        115728 non-null  object
 27  payment_sequential             115728 non-null  int64
 28  payment_type                   115728 non-null  object
 29  payment_installments           115728 non-null  int64
 30  payment_value                  115728 non-null  float64
 31  product_category_name          114090 non-null  object
 32  product_name_lenght            114090 non-null  float64
 33  product_description_lenght     114090 non-null  float64
 34  product_photos_qty             114090 non-null  float64
 35  product_weight_g               115708 non-null  float64
 36  volume_cm3                     115708 non-null  float64
dtypes: float64(8), int64(6), object(23)
memory usage: 33.6+ MB
```

Data numérique

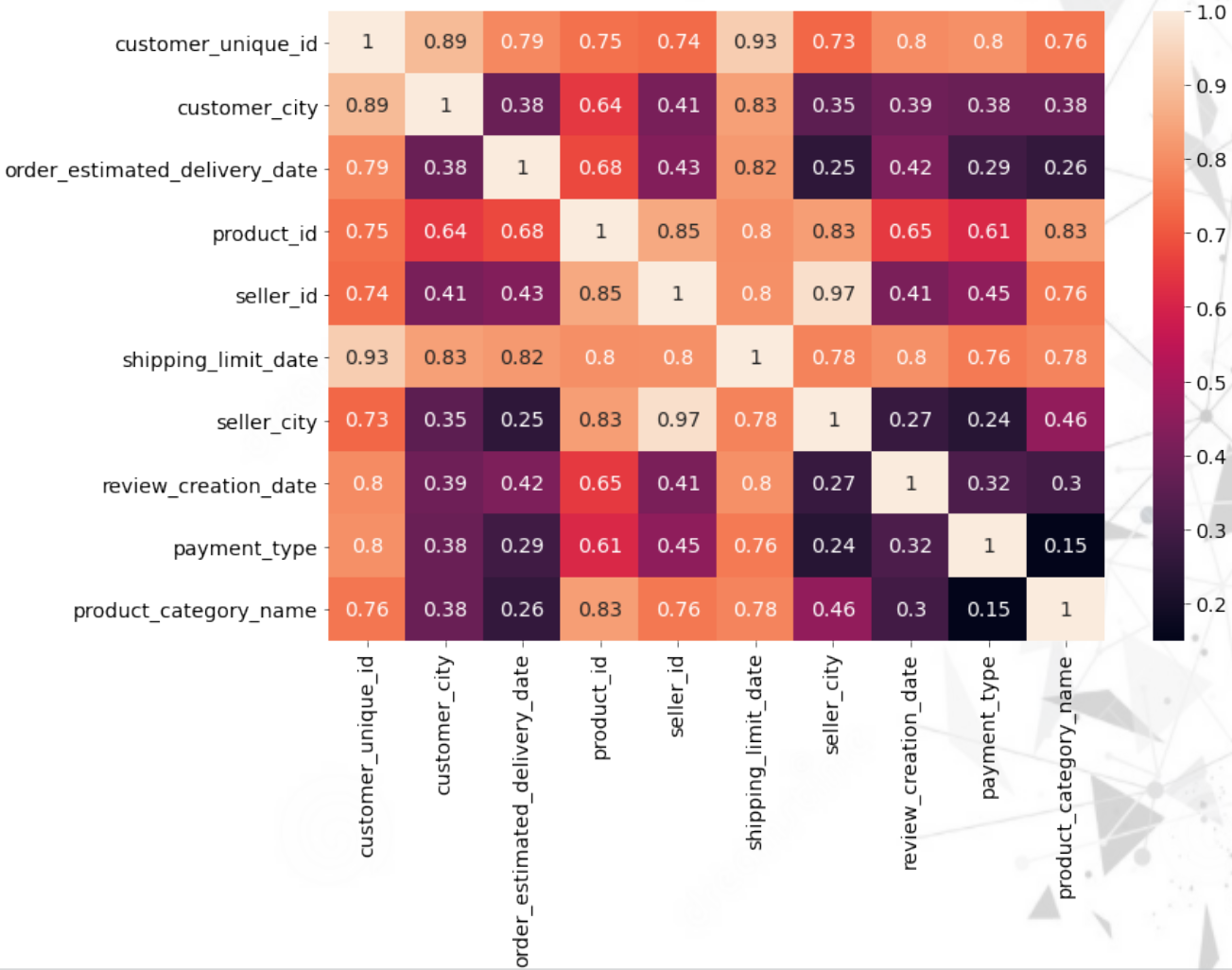


Data cattégorielle



Data cattégorielle

Nous appliquons *Cramers V statistiques* sur nos varibales catégorielle du Dataframe. Ce code est crée et publié par "Wicher Bergsma":



Plan

Duplication de lignes no dupliquées

```
In [70]: data_2018[data_2018['customer_unique_id']=='31318a0597cd9d50ce4cfd03c80fe780'].T
```

Out[70]:

	9	10
customer_unique_id	31318a0597cd9d50ce4cfd03c80fe780	31318a0597cd9d50ce4cfd03c80fe780
customer_zip_code_prefix	37540	37540
order_estimated_delivery_date	2018-03-12 00:00:00	2018-03-12 00:00:00
order_item_id	1	2
product_id	a9516a079e37a9c9c36b9b78b10169e8	a9516a079e37a9c9c36b9b78b10169e8
seller_id	7c67e1448b00f6e969d365cea6b010ab	7c67e1448b00f6e969d365cea6b010ab
shipping_limit_date	2018-02-13 03:47:31	2018-02-13 03:47:31
price	106.99	106.99
freight_value	21.76	21.76
seller_zip_code_prefix	8577	8577
review_score	2	2
payment_sequential	1	1
payment_type	boleto	boleto
payment_value	257.5	257.5
product_category_name	moveis_escritorio	moveis_escritorio

Duplication de lignes no dupliquées

Cette situation provoquera des résultats irréalistes. Nous nous débarrassons des "lignes dupliquées" qu'un client peut avoir s'il a acheté l'article plus q'une fois par commande, ou s'il a payé l'article acheté en plusieurs fois.

De cette manière, nous évitons de nous répéter et de compter les clients ou les produits/prix plus que la situation réelle.

Plan

Un coup d'œil sur les critères de clustering en marketing:

1. Segmentation comportementaux (*nombre de visites, paiement, ..*).
2. Segmentation géographique (*Géolocation, météo, ..*).
3. Segmentation psychographique (*intérêt, perso, ..*).
4. Segmentation démographique (*genre, revenue, ..*).

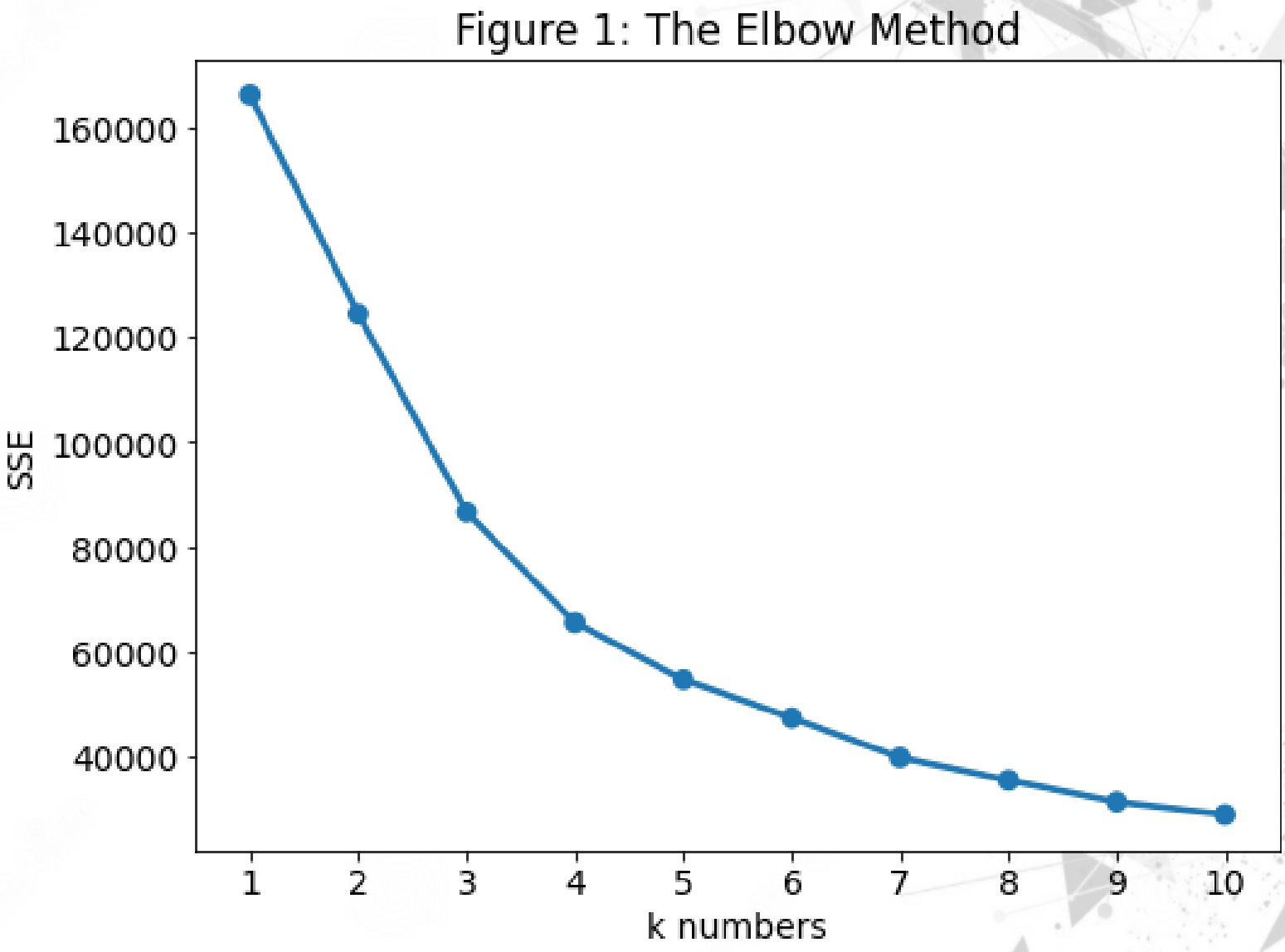
Les modèles de clustering:

1. Segmentation comportementaux, nous appliquons les modèles **KMeans** et **Agglomerative.**
2. Segmentation géographique, nous appliquons *t-SNE* suivie par les modèles **MiniBatchKMeans** et **Birch**.
3. Segmentation psychographique, nous appliquons *UMAP* suivie par **GaussianMixture**.

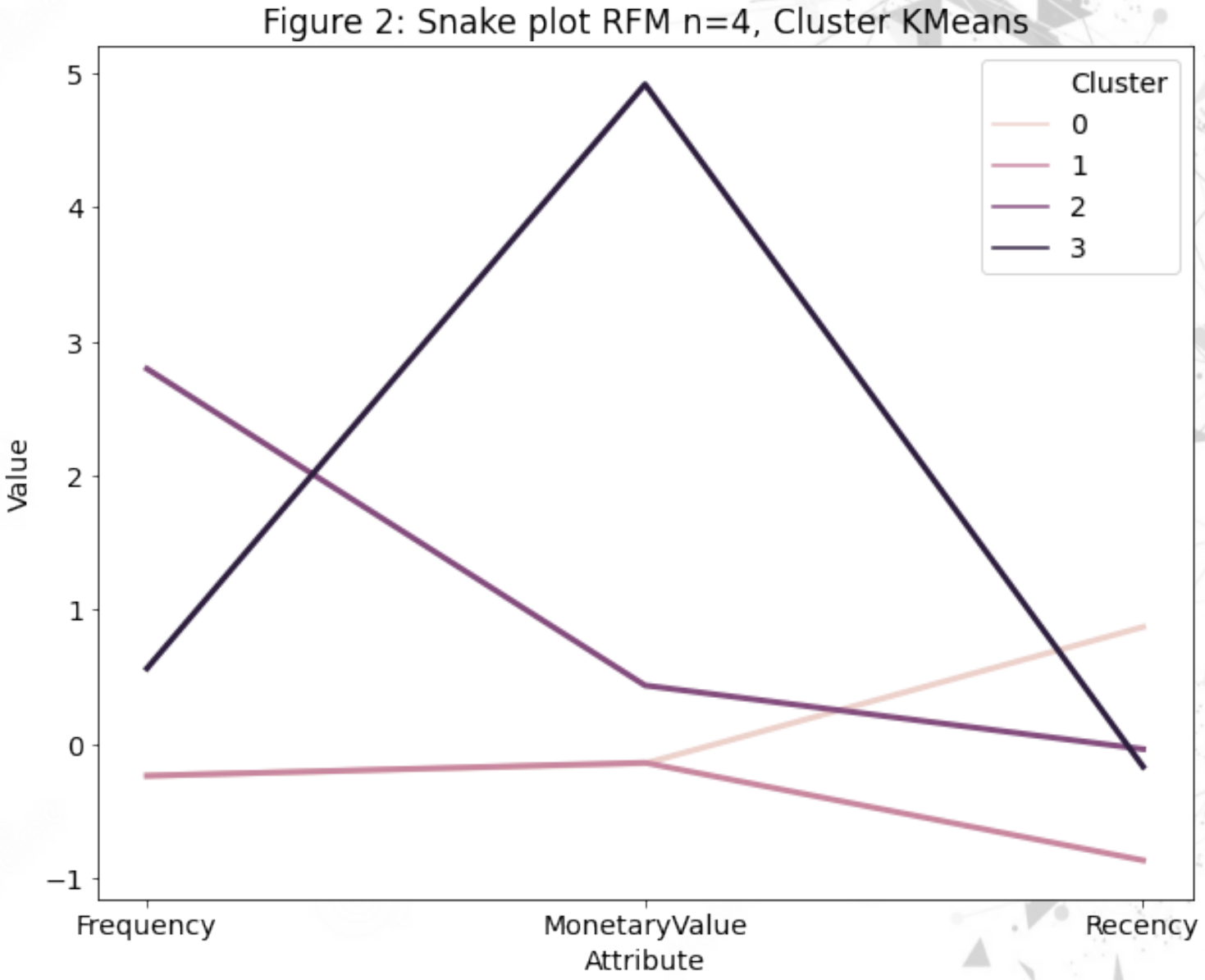
La stratégie suivie pour le choix des hyperparamètres et l'évaluation des performances:

1. Pour le RFM, nous appliquons la méthode **Elbow method** au KMeans, et **dendrogram** pour l'algorithme Agglomerative:
   - 1.1 metric silhouette_score()
   - 1.2 metric davies_bouldin_score()
   - 1.3 metric Variance Ratio Criterion
2. Pour Géolocation clustering, nous appliquons **train/test split** sur les deux méthodes MiniBatchKMeans et Birch:
   - 2.1 metric homogeneity_completeness_v_measure()
3. Pour psychographique clustering, nous appliquons **GaussianMixture.bic()** en cross validation:
   - 3.1 metric homogeneity_completeness_v_measure()

Clustering Machine Learning – RFM

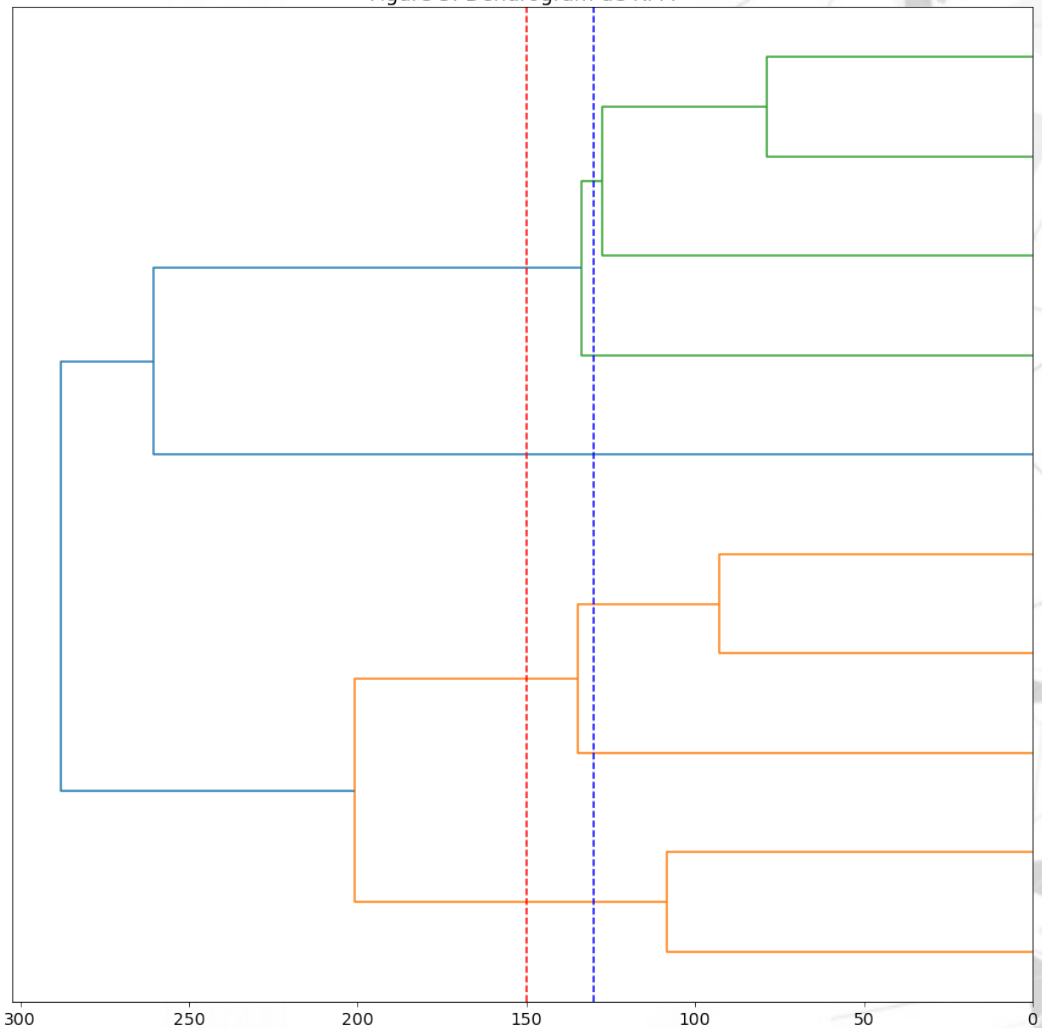


Clustering Machine Learning – RFM

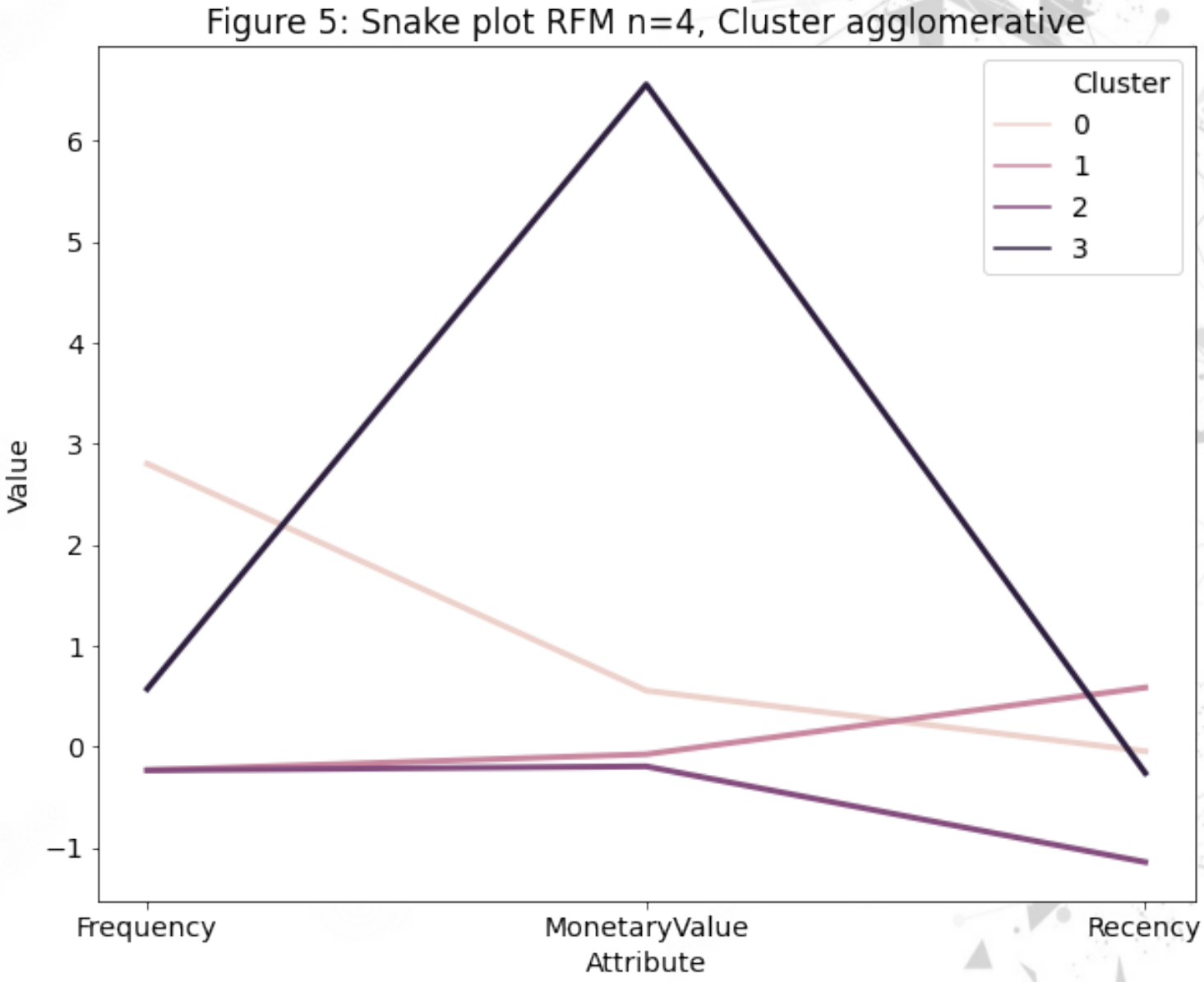


Clustering Machine Learning – RFM

Figure 3: Dendrogram de RFM

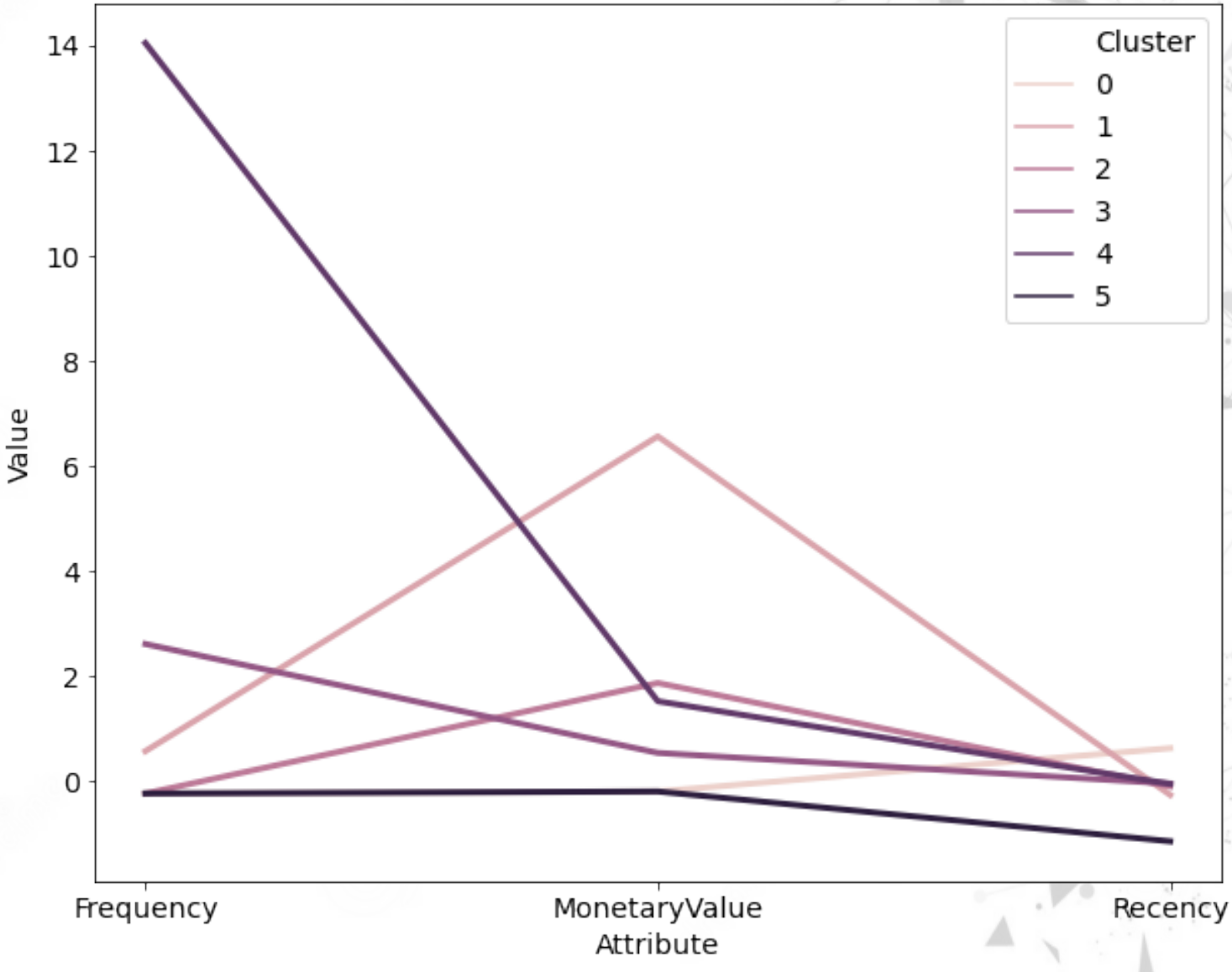


Clustering Machine Learning – RFM



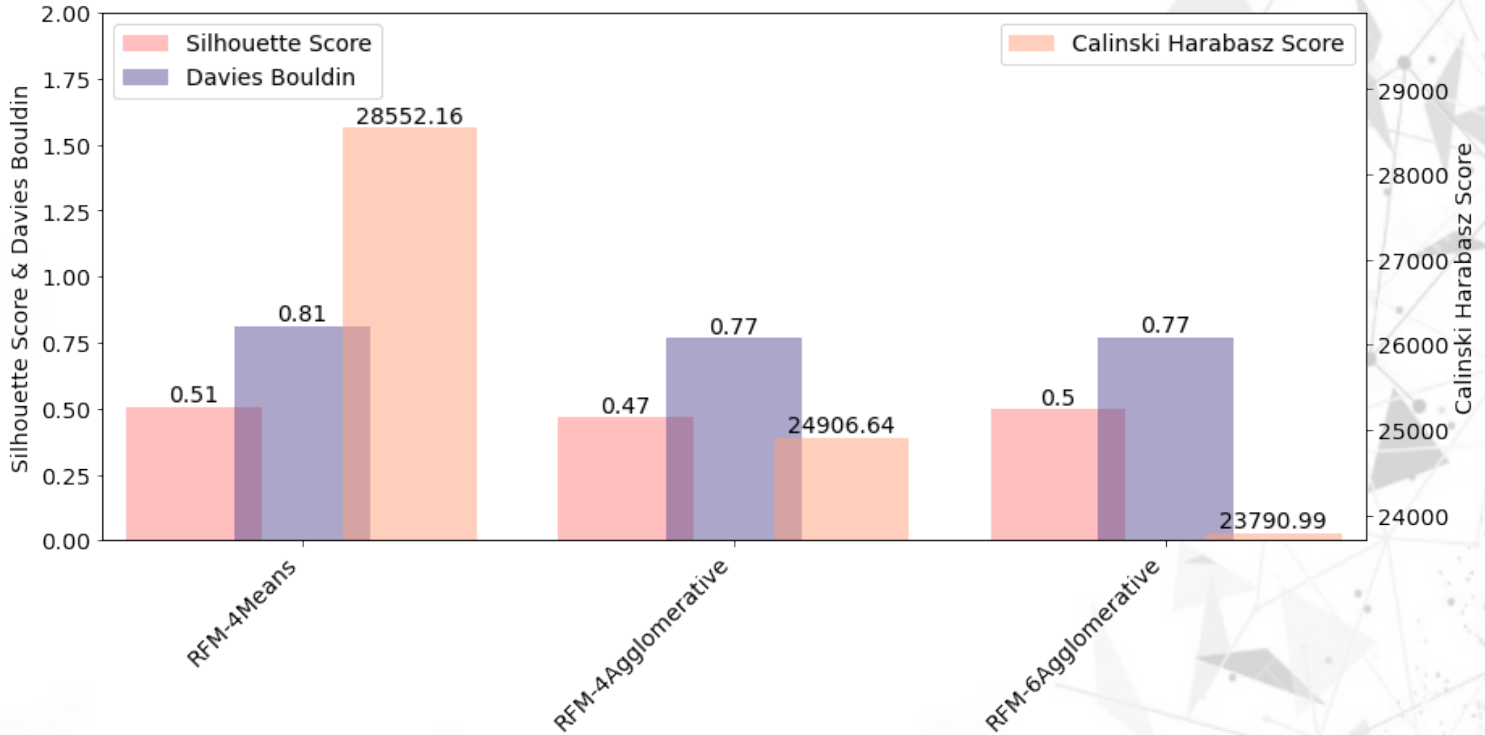
Clustering Machine Learning – RFM

Figure 6: Snake plot RFM n=6, Cluster agglomerative



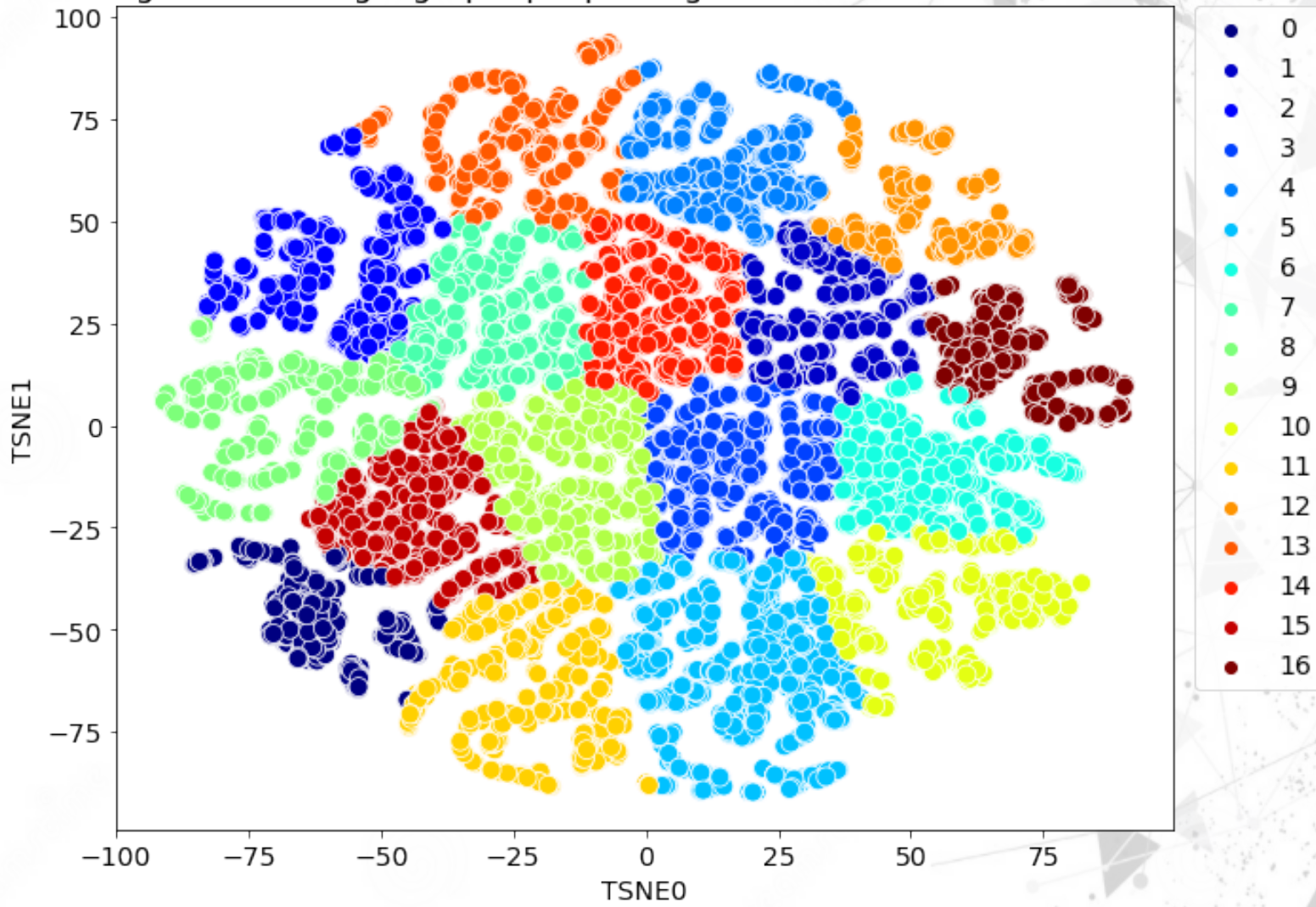
Clustering Machine Learning – RFM Comparaison

La comparaison se fera cluster par cluster. Voiçi les résultats de la comparaison RFM:



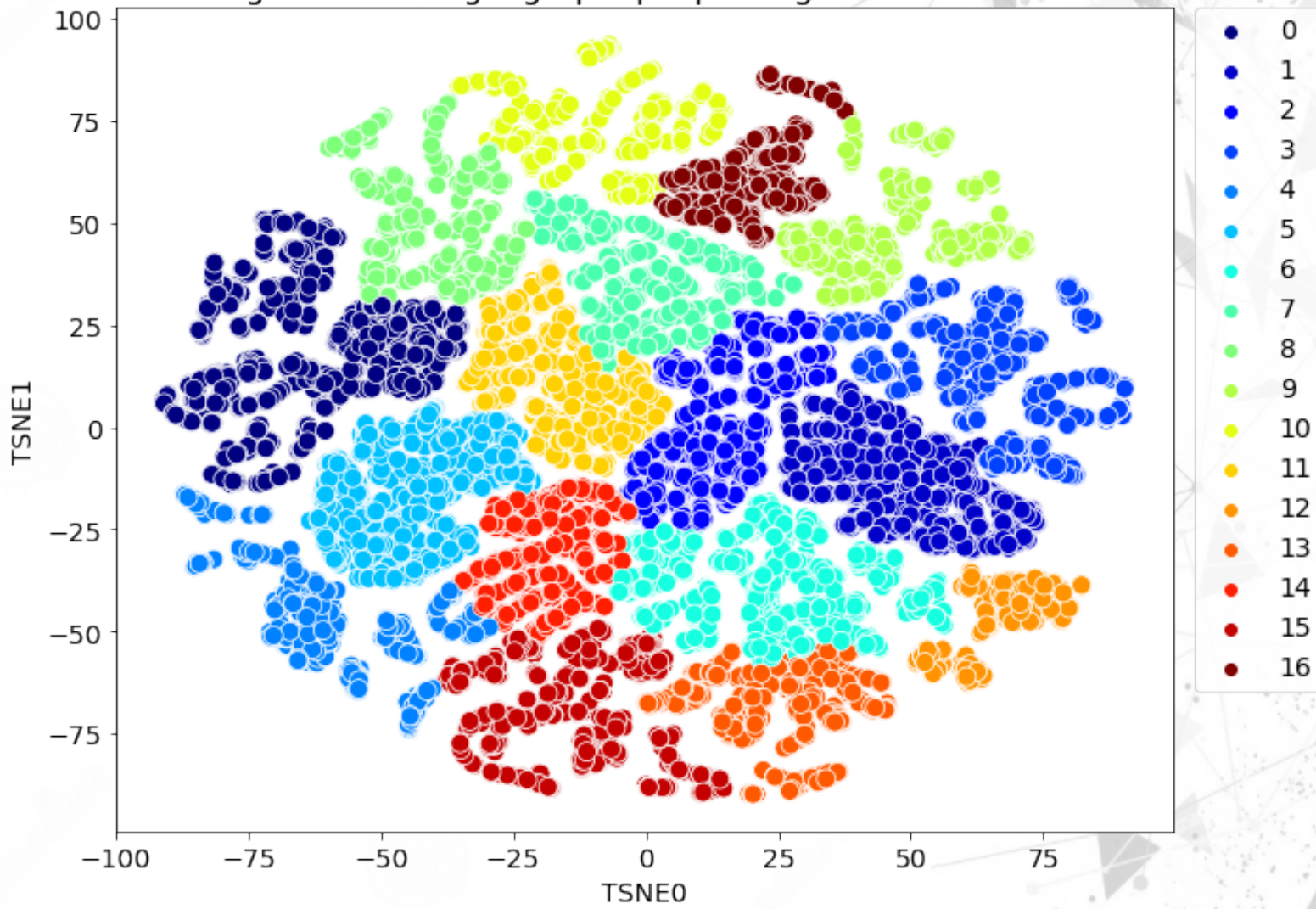
Clustering Machine Learning – Géolocation

Figure 7: TSNE géographique par segmentation Mini Barch KMeans



Clustering Machine Learning – Géolocation

Figure 8: TSNE géographique par segmentation Birch

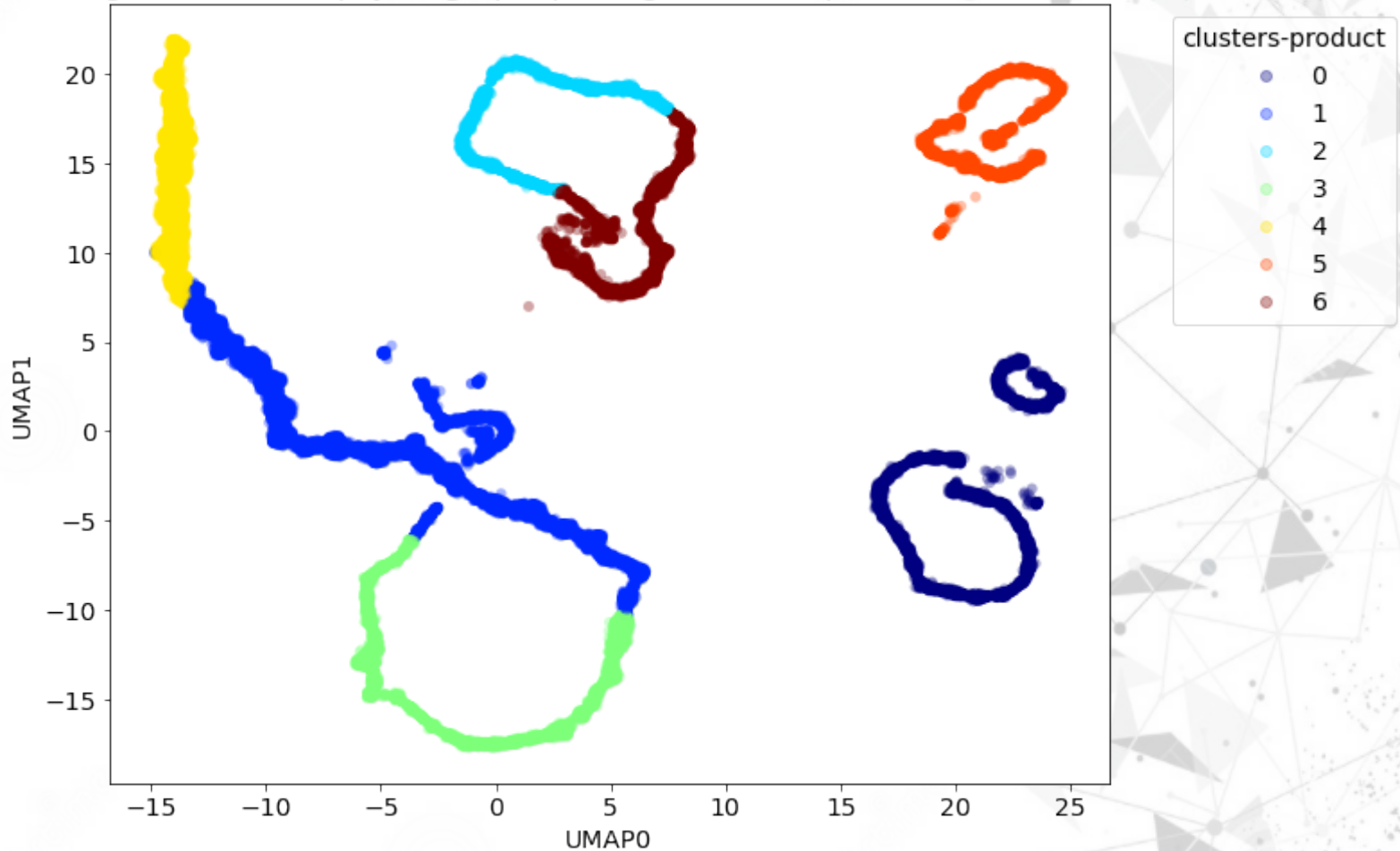


La comparaison se fera cluster par cluster. Voiçi les résultats de la comparaison Géolocation:

	BIRCH	Mini-KMeans
homogeneity	0.265440	0.284029
completeness	0.973019	0.974140
v_measure	0.417096	0.439820

Clustering Machine Learning – psychographique

Figure 10: UMAP psychographique segmentation par GaussianMixture

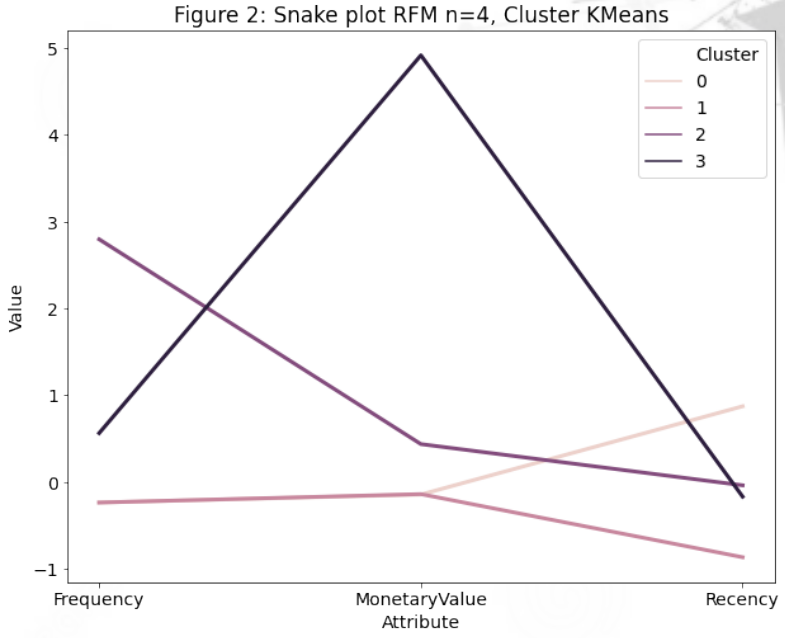


La comparaison se fera cluster par cluster. Voiçi les résultats de la comparaison psychographique:

	GaussianMixture
homogeneity	0.017843
completeness	0.031561
v_measure	0.022797

Plan

Le choix finale et le contrat du maintenance



1. Cluster 1 sont les clients pas fidèles.
2. Cluster 0 représente les nouveaux clients, un suivi ciblé peut les convertir en clients réguliers.
3. Cluster 2 ont été autrefois des clients fidèles mais ont cessé de l'être depuis. Un message ciblé peut les réactiver.
4. Cluster 3 représente les clients qui ont souvent acheté et dépensé des sommes importantes, mais qui n'ont pas acheté récemment. Envoyez-leur des campagnes de réactivation personnalisées pour renouer le contact.

Plan

Conclusion & perspectives

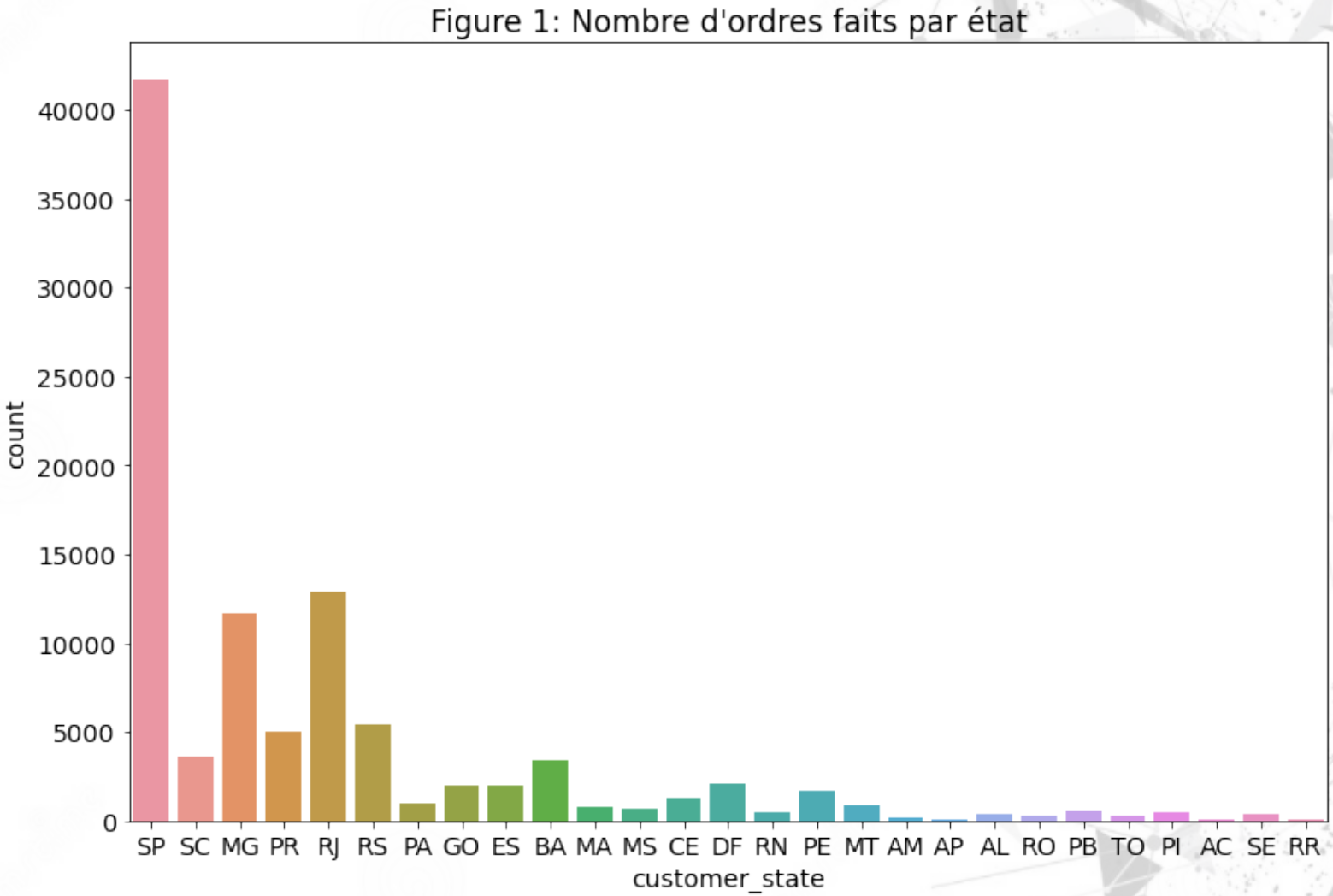
Dans un premier temps, nous avons:

- testé plusieurs méthodes de clustering pour segmenter les clients d'Olist.
- testé plusieurs critères de clustering, RFM, Géolocation, Psychographique.
- présenté plusieurs possibilités pour gérer le cluster: Hyperparamétres, cross validation, train/test split, labels.
- utilisé une "Distribution based method", "Centroid based clustering", "Density based clustering" and "hierarchical clustering".

Dans un deuxième temps, nous pouvons:

- recalculer RFM KMeans avec la répartition train/test et tester à nouveau avec d'autres méthodes de clustering.
- aussi ne pas être limités par le clustering par Machine Learning, consultez la section suivante.

Clustering sans Machine Learning



Clustering sans Machine Learning