Segmentez les clients d'un site e-commerce

L.khellouf[1] Mentor: B.Beaufils[2]

[12]OpenClassRooms

Projet 4, Avril 2021

Sommaire

Problématique

Olist, solution de vente sur les marketplaces en ligne, souhaite segmenter ses clients pour rendre plus efficaces leurs campagnes de communication.

Objectifs:

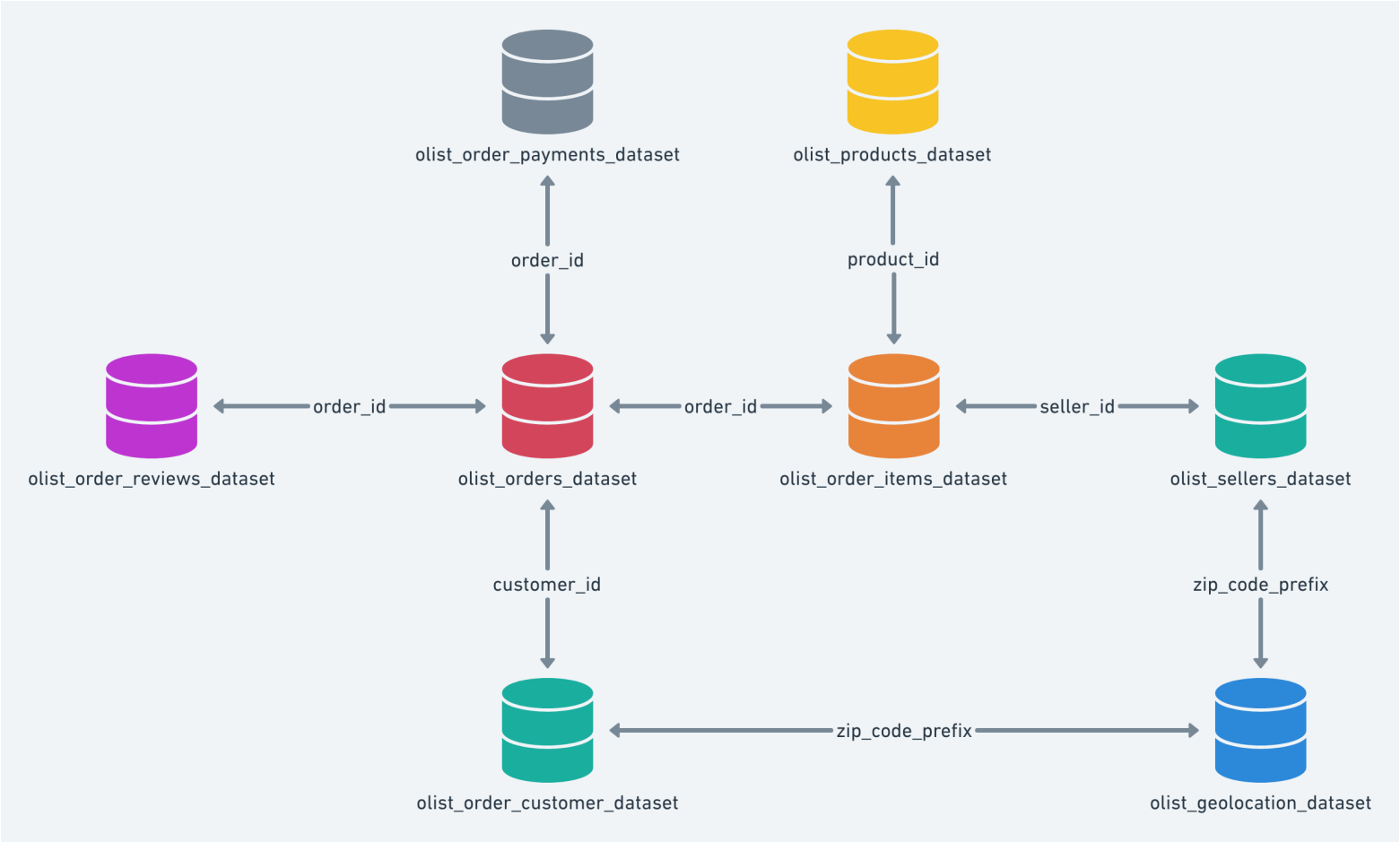
Comprendre les différents types d'utilisateurs grace à leur comportement et à leur données personnelles.

Mission:

- La segmentation proposée doit être exploitable facile d'utilisation pour l'équipe marketing.
- Evaluer la fréquence à laquelle la segmentation doit être mise à jour, afin de pouvoir effectuer un devis de contrat de maintenance.
- Le code fourni doit respecter la convention PEP8, pour être utilisable par Olist.

Présentation des données:

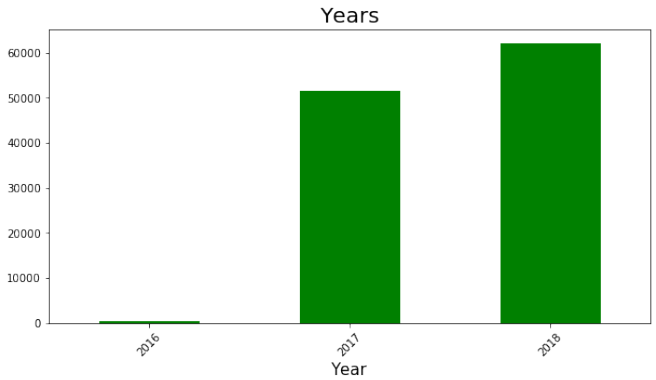
- Les données se réfèrent au commerce électronique sur le territoire brésilien entre 09/2016 et 10/2018. Et ils sont disponibles sur: https://www.kaggle.com/olistbr/brazilian-ecommerce

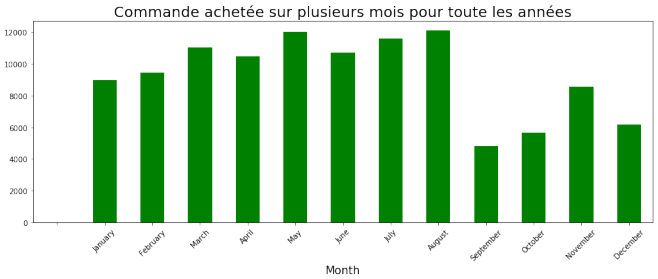


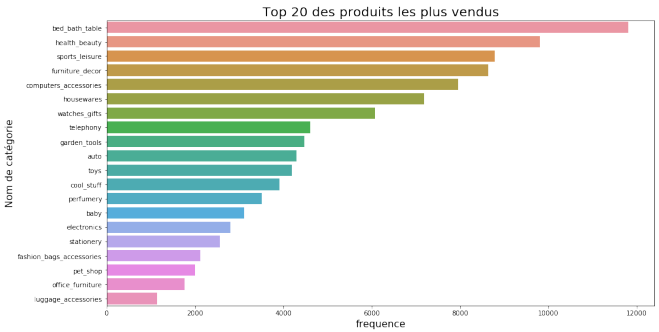
Découverte des données:

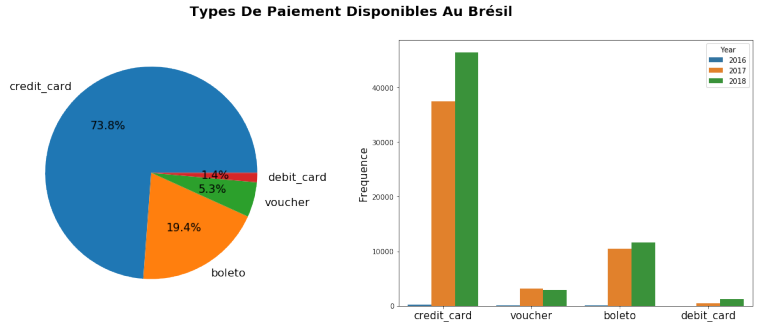
	data	colums	colums_nbr	null_nbr	null_colums_nbr	null_colums
0	customers	customer_id,customer_unique_id,customer_zip_code_prefix,customer_city,customer_state	5	0	0	
1	geolocation	geolocation_zip_code_prefix,geolocation_lat,geolocation_lng,geolocation_city,geolocation_state	5	0	0	
2	order_items	order_id,order_item_id,product_id,seller_id,shipping_limit_date,price,freight_value	7	0	0	
3	order_payments	order_id,payment_sequential,payment_type,payment_installments,payment_value	5	0	0	
4	order_reviews	review_id,order_id,review_score,review_comment_title,review_comment_message,review_creation_date,review_answer_timestamp	7	146532	2	review_comment_title, review_comment_message
5	orders	order_id,customer_id,order_status,order_purchase_timestamp,order_approved_at,order_delivered_carrier_date,order_delivered_customer_date,order_estimated_delivery_date	8	4908	3	order_approved_at, order_delivered_carrier_date, order_delivered_customer_date
6	products	product_id,product_category_name,product_name_lenght,product_description_lenght,product_photos_qty,product_weight_g,product_length_cm,product_height_cm,product_width_cm	9	2448	8	product_category_name, product_name_lenght, product_description_lenght, product_photos_qty, product_weight_g, product_length_cm, product_height_cm, product_width_cm
7	sellers	seller_id,seller_zip_code_prefix,seller_city,seller_state	4	0	0	
8	product_category_name	product_category_name,product_category_name_english	2	0	0	

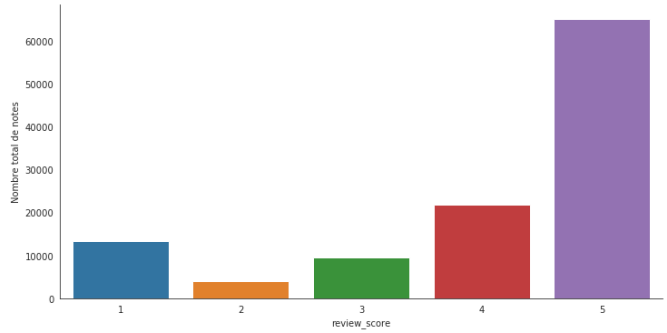
Analyse des indicateurs

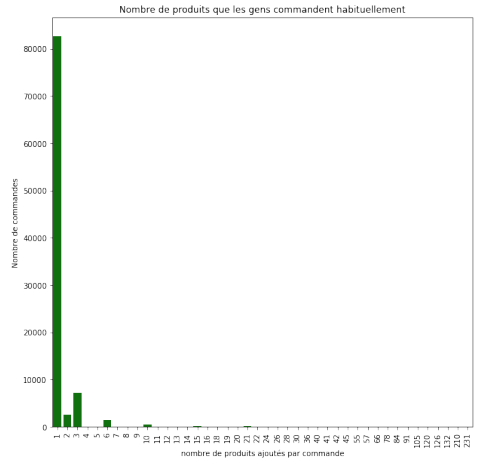






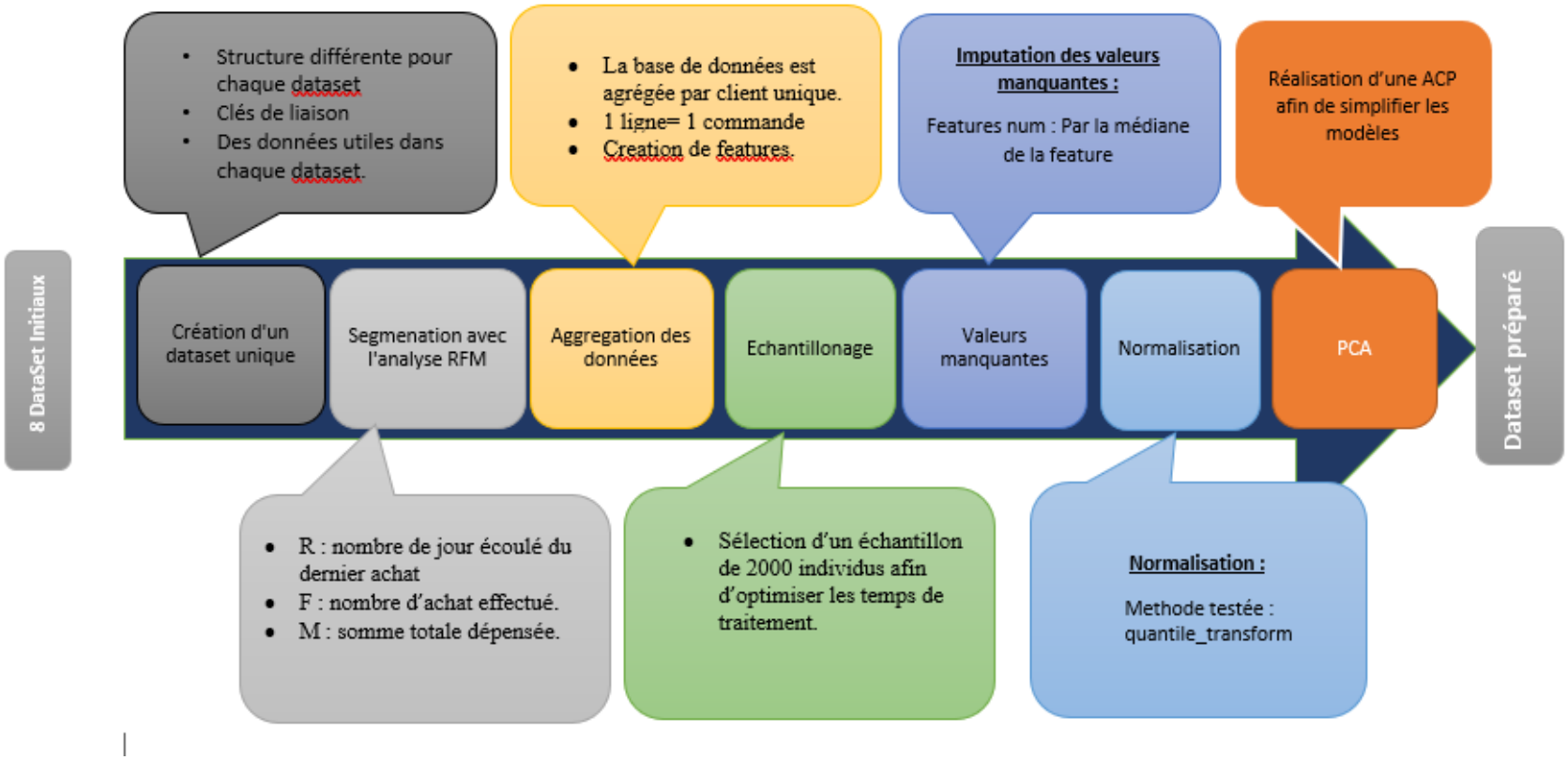




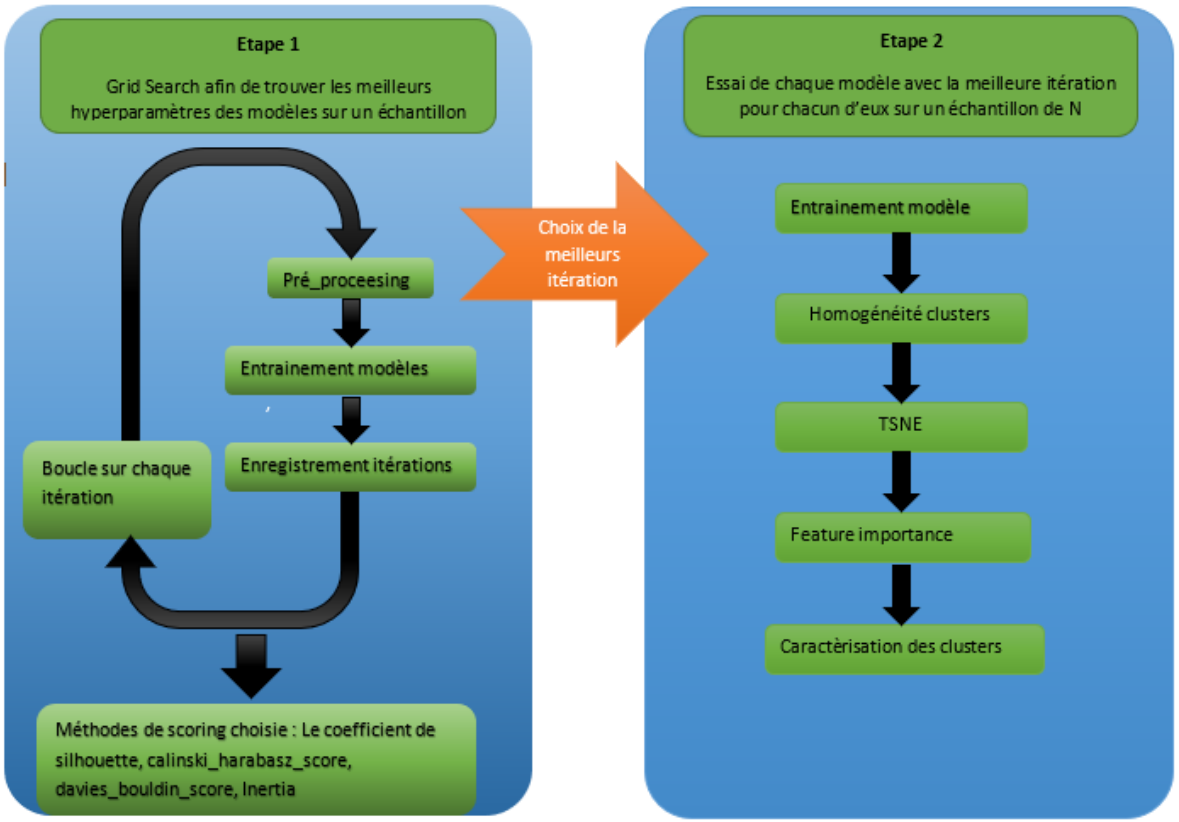


Feature Engeneering

Pipeline de Preprocessing:



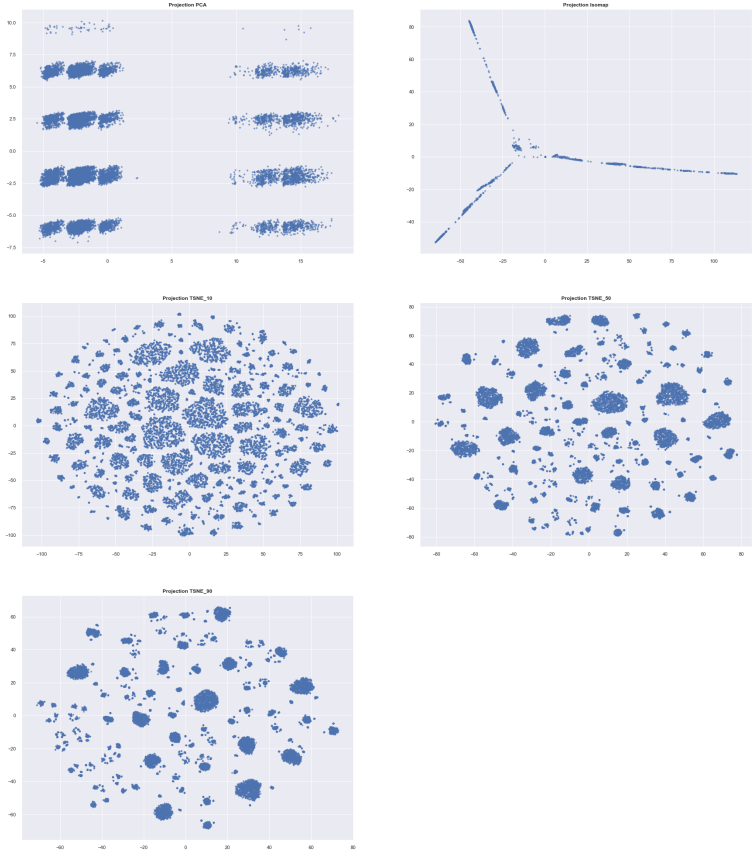
Modélisation



Analyse Etape 1

Methodes de réduction diméntionnelle

Test de plusieurs méthodes de réduction dimensionnelle

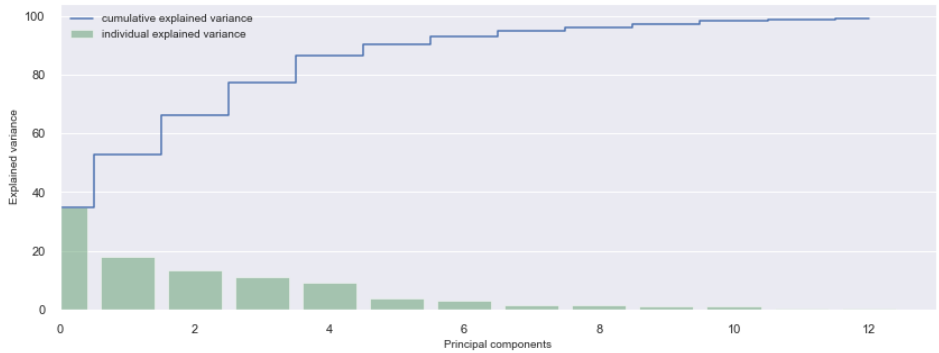


La PCA a été retenu par rapport aux autres méthodes pour sa capacité à projeter des clusters denses et séparés (non encore labellisés à ce stade) et à sa rapidité.

Les variables utiliser sont numérique car agréger par clients uniques implique des problèmes sur les variables catégorielles. Il n'y a aucun sens à effectuer la somme ou la moyennes sur une variable de type catégorie de produit par exemple.

PCA:

99 % d'informations concervées. Réduction de features à 12.



La modélisation:

Trois modèles de clustering ont été testés: K-Means, DBSACN et Agglomerative Clustering. La détermination du nombre de clusters optimal a été faite à partir de la visualisation de l'evolution de métriques propres aux problématiques de clustering en fonction de n_cluster. Les métriques utilisées sont l'inertie, le score de Silhouette, le coefficient de Calinski Harabasz et le coefficient de Bouldin. D'après les résultats 6 clusters sont à utiliser.

Analyse Etape 1

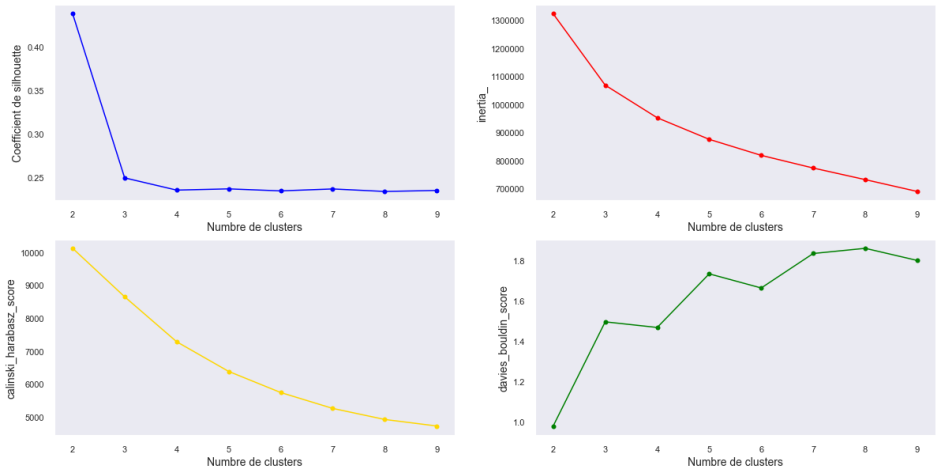
k-means

L'algorithme de K_Means est utilisé pour le clustering des variables numériques. L'objectif du clustering K-Means est de minimiser la variance intra-cluster totale

Algorithme 1 Algorithme K-Means

1: Choisir aléatoirement K points(centre de gravité) de l'hyper-plan(dimension correspondant au nombre de features)
2: Calculer la distance euclidienne de chaque observation aux centres de gravité
3: Affecter chaque observation au centroïde de distance minimale
4: Mise à jour des centroïdes par la moyenne des observations de leur cluster

Le Grid Search:
- n_ clusters= nombre k de clusters [2:10]
- n_init= nombre d'exécustion [5, 10, 20]
- max_iter = nombre d'itérations [50, 100, 300]
- Init= type d'initialisation [Random, k_means++]

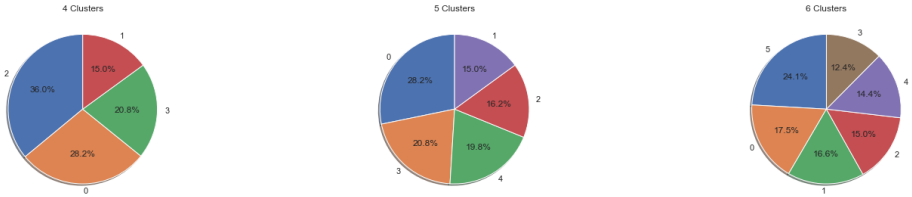


Les 4 graphiques ci-dessus permettent de visualiser l'évolution de chaque métrique en fonction du nombre de clusters. Chaque plot indique un nombre optimale de clusters différents. Ce nombre semble être 2. Cependant deux paraît assez faible. Pour certains plots un changement dans l'évolution des métriques est observé à $n_cluster = 6$.

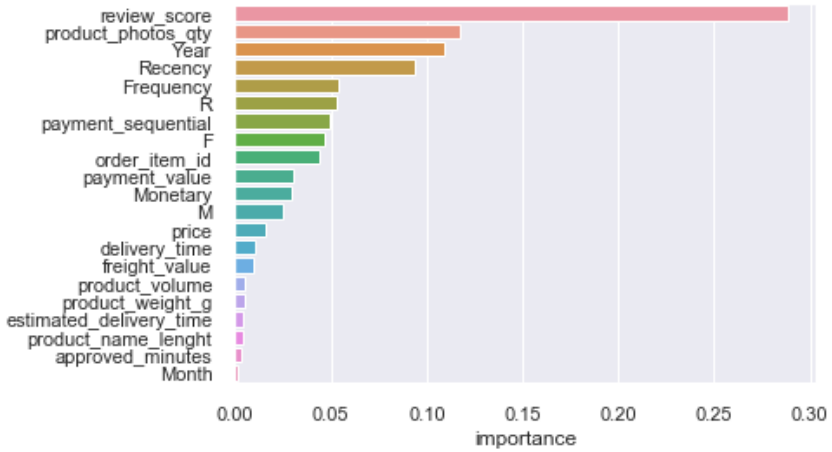
Analyse Etape 1+ 2

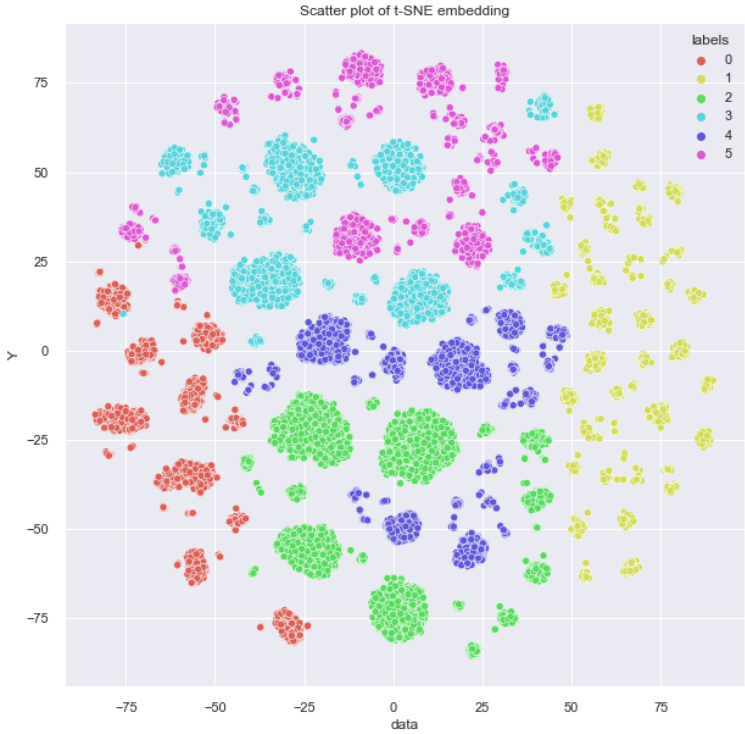
k-means

Analyses résultats: Nombre de clusters raisonnable avec une bonne homogénéité générale



On cherche à savoir l'importance relative de chaque feature à la prédiction de l'étiquette du cluster issu d'un modèle de classification "RandomForestClassifier"

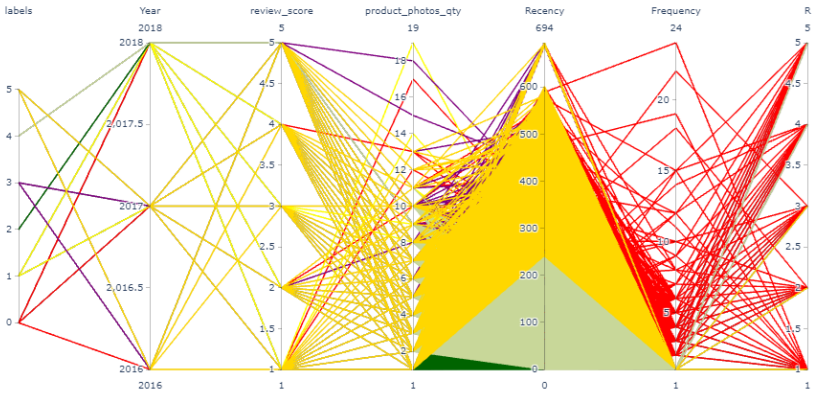




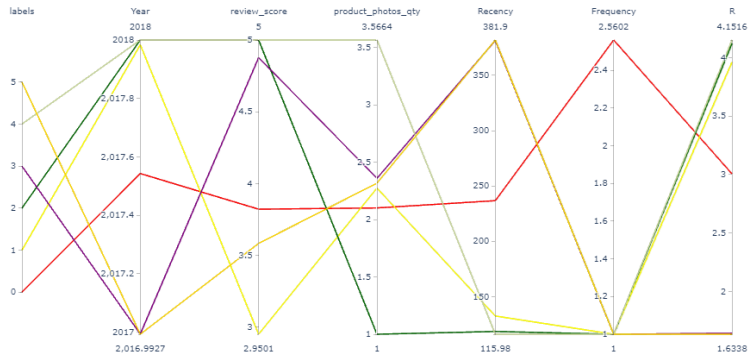
Analyse des résultats:

Graph_ Object

Pour la caractérisation de nos 6 clusters, nous avons utiliser la méthode de features_importance associée au classifier Random Forest et ceci a permis de faire ressortir 5 variables sur les 21 initiales.

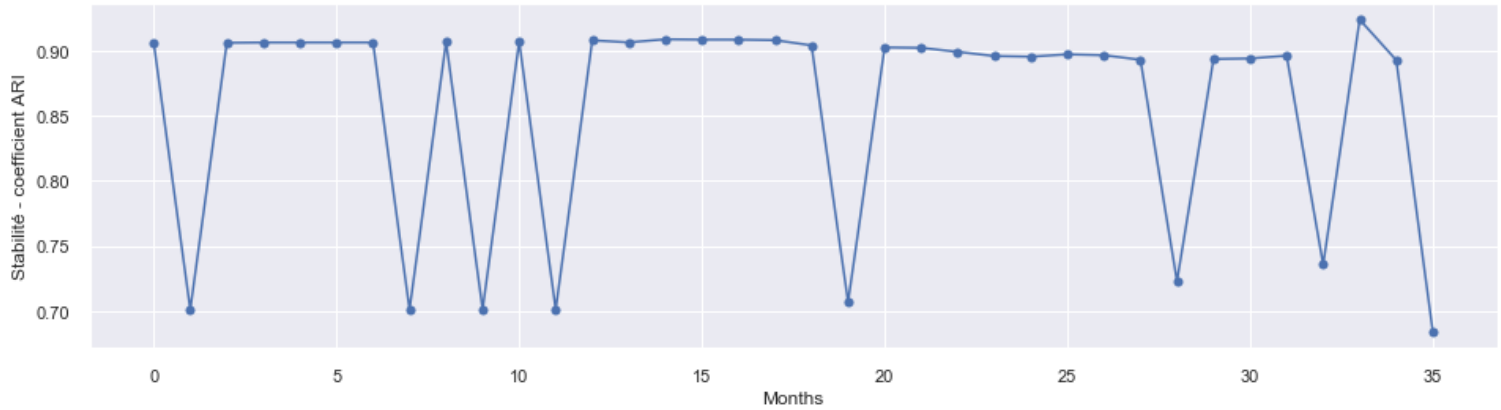


Les valeurs moyennes par variables et par clusters



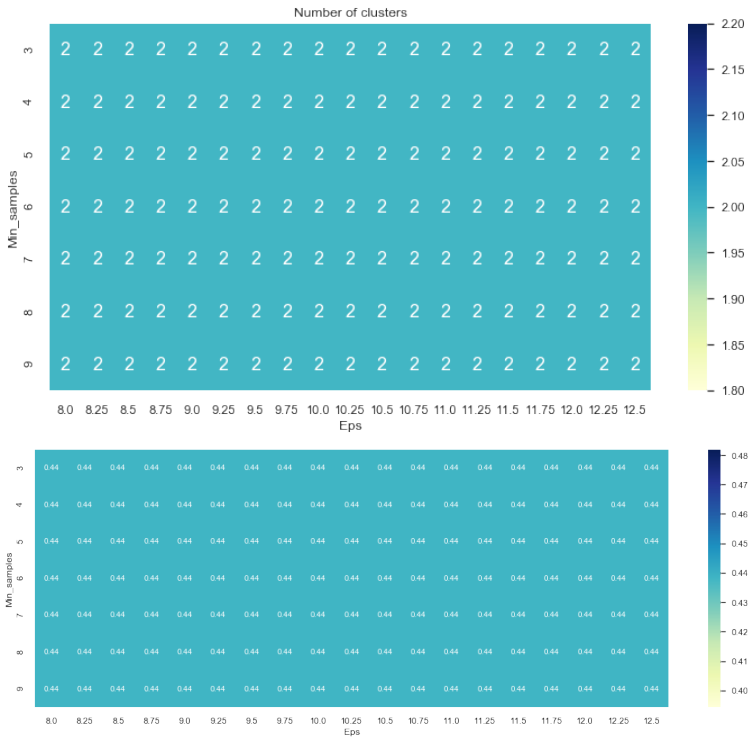
Stabilité temporelle

Pour la maintenance nous avons analysé la stabilité temporelle de l'attribution des labels de clusters dans le temps pour chaque mois de la base de données clients. Les résultats obtenus ont été plottés pour visualiser l'évolution temporelle. La stabilité chute le 2, 7, 9, 11, 19, 28, 32, 35 mois.Sinon elle reste stable (ARI= 0.90).

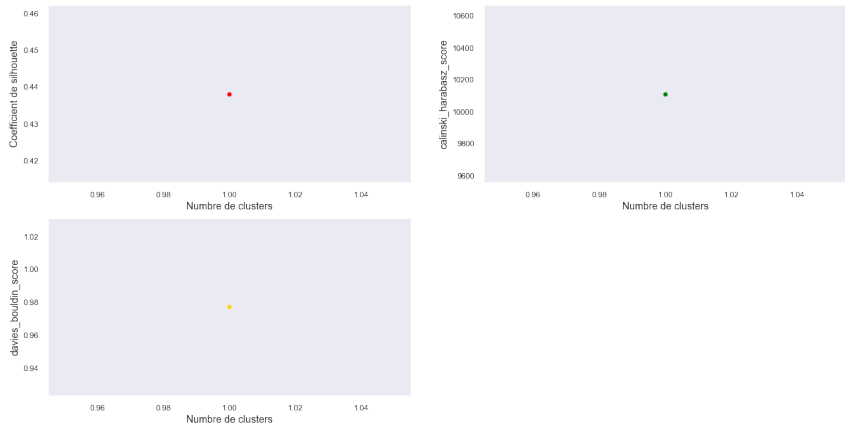


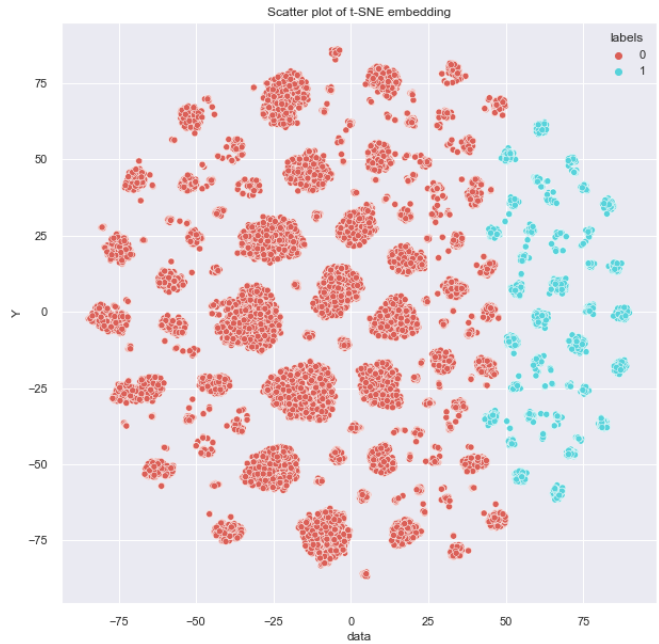
DBSACN

- L'idée centrale de DBSCAN est autour du concept de régions denses. L'hypothèse est que les clusters naturelles sont composées de points densément localisés.
- Parmi les avantages (resp: les incovients) de DBSCAN on a : Elle trouve elle meme le nombre de clusters, basé sur les paramètres Eps et MinPts et elle détecte les valeurs abérantes(resp: Temps d'éxécution élevées à O(n log (n)))



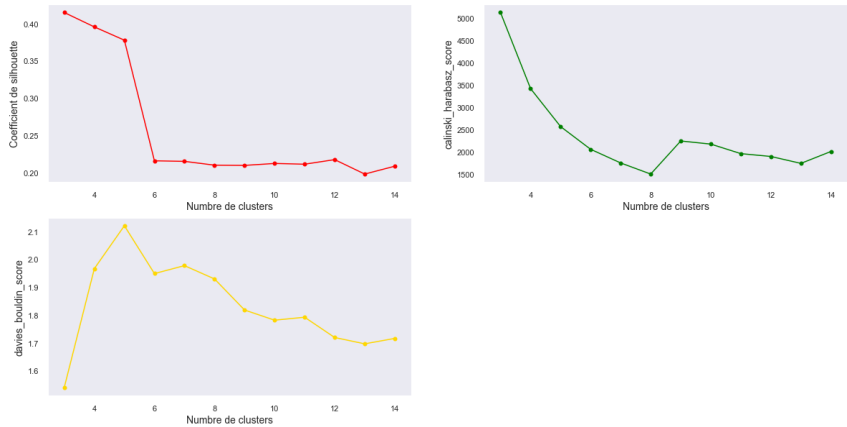
Le Grid Search:
- eps_values = np.arange(8,12.75,0.25)
- min_samples = np.arange(3,10)





AgglomerativeClustering:

Le Grid Search:
- n_clusters= range(3, 15)
- clustering_algorithms = ('single','average', 'complete', 'ward')



- Ce mécanisme de clustering trouve les points de données les plus proches les uns des autres et les regroupe successivement. - Le clustering agglomératif est hiérarchique car il effectue des opérations de manière séquentielle. Cet algorithme est utile dans les cas où on souhaite prendre des décisions sur la façon dont on souhaite regrouper nos données de manière grossière ou fine, ou dans quelle résolution on veut pour nos données.

Algorithm of Agglomerative Clustering

1. Make each data point as a single-point cluster.
2. Take the two closest distance clusters by single linkage method and make them one clusters.
3. Repeat step 2 until there is only one cluster.
4. Create a Dendrogram to visualize the history of groupings.
5. Find optimal number of clusters from Dendrogram.

Conclusion

MODELISATION
A partir de 8 datasets : Création d'un dataset listant les détails de commandes
Entraînement de trois modèles de classification non supervisée : K-means et CAH ont des résultats similaires. Cependant, le temps de traitement du CAH est très long contrairement au K-means. C'est pourquoi nous préfèrerons le K-means. DBSCAN, est un modèle très compliqué à régler. 6 clusters exploitables facilement.

Merci pour votre attention