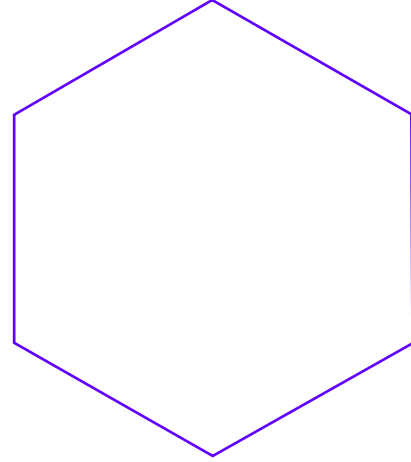


Segmentation des clients d'Olist



Objectifs



- Effectuer une segmentation et une analyse des profils des clients du site d'e-commerce *Olist*.
- Suggérer un contrat de maintenance fondé sur l'analyse des profils des clients.
- Les données ont été collectées entre 2016 et la fin de 2018.
- Période d'analyse sélectionnée : 04-09-2016 au 17-10-2018 (commandes passées par les clients)



Requêtes SQL



Commandes récentes (- 3mois)
ayant eu un retard de livraison d'au
moins 3 jours

444 commandes avec un retard d'au
moins 3 jours



3 vendeurs de moins de 3 mois
ont vendus plus de 30 produits



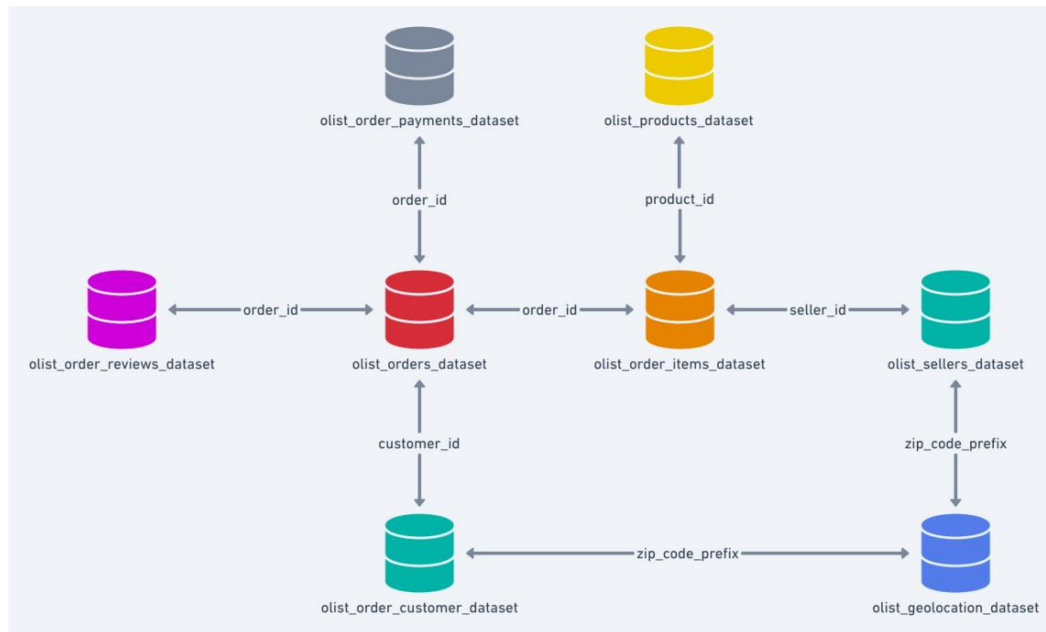
18 vendeurs avec un CA de plus de
100,000 reals



Les états avec les plus
mauvaises notes moyennes
depuis un an sont **Santana de
Parnaíba, rio de Janeiro,
Campinas, Congonhas et Sao
Paulo.**

Analyse exploratoire - SQL

Exploration des 9 bases de données suivantes
(+ base de données translation)



```
--Nombre de clients par ville--  
SELECT c.customer_city, COUNT(c.customer_id) AS customer_by_city  
FROM customers c  
GROUP BY customer_city  
ORDER BY customer_by_city DESC ;  
--16% des clients sont situés à Sao Paulo--
```

customer_city	customer_by_city
sao paulo	15 540
rio de janeiro	6 882
belo horizonte	2 773
brasilia	2 131
curitiba	1 521
campinas	1 444
porto alegre	1 379
salvador	1 245
guarulhos	1 189

Par exemple analyse de la répartition clients selon la ville: 16% viennent de Sao Paulo, 7% de Rio de Janeiro, 2% de Brasilia etc.

Création de la base finale - SQL

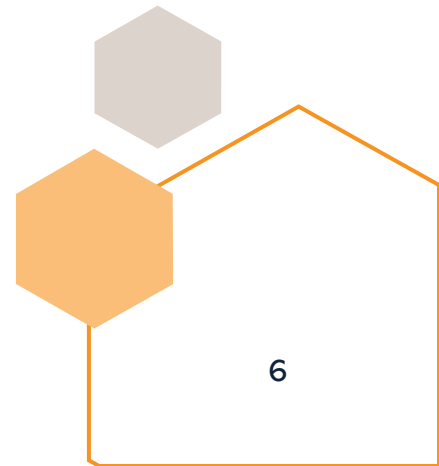
Toutes les jointures sont des *LEFT JOIN* afin de garder toutes les informations de la base initiale (base de gauche). Base finale avec 119,143 lignes représentant un article par commande:

- ➡ Jointure entre la table « *customer* » et « *orders* » sur « *customer_id* » : **Objectif** d'associer chaque client à ses commandes. Chaque ligne représente une commande , cela nous permet de garder les clients qui ont effectué plusieurs commandes.
- ➡ Jointure avec la table « *order_items* » sur « *order_id* » : **Objectif** est d'ajouter les informations sur les articles de la commande. Si une commande comporte plusieurs articles, chaque article créera une nouvelle ligne
- ➡ Jointure avec la table « *order_pymts* » sur « *order_id* » : **Objectif** est d'ajouter les informations de paiements de chaque commande. Une commande peut avoir plusieurs types de paiements et plusieurs échelonnages.
- ➡ Jointure avec la table « *order_review* » sur « *order_id* » : **Objectif** est d'ajouter les avis clients à chaque commande. Certaines commandes peuvent ne pas avoir d'avis, LEFT JOIN permet de garder toutes les commandes même celles sans avis.
- ➡ Jointure avec la table « *products* » sur « *product_id* » : **Objectif** est d'ajouter les informations sur l'article de chaque commande, en particulier sa catégorie.
- ➡ Jointure avec la table « *translation* » sur « *product_id* » : **Objectif** est d'ajouter les traductions de catégories de chaque article
- ➡ Jointure avec la table « *sellers* » sur « *sellers_id* » : **Objectif** est d'ajouter les informations sur les vendeurs.

Création des variables RFM - SQL

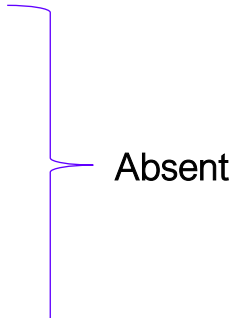
- R pour « *Recency* » : nombre de jours écoulés depuis la dernière commande. La date de référence est la dernière date de commande de toute la base (17-10-2018) et j'effectue une soustraction entre la date de référence et la dernière date de commande pour chaque client.
- F pour « *Frequency* » : nombre total de commande passée par client sur la période d'analyse
- M pour « *Monetary* » : valeur totale dépensée par client sur toute la période d'analyse
- Ici, une « *INNER JOIN* » est effectuée pour s'assurer de garder seulement les clients qui ont effectué au moins une commande.

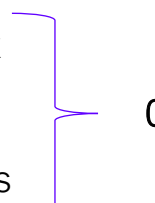
```
●CREATE TABLE final_table_rfm AS
WITH rfm AS (
  SELECT
    customer_unique_id,
    CAST(julianday('2018-10-17') - julianday(MAX(order_purchase_timestamp)) AS INTEGER) AS recency,
    CAST(COUNT(DISTINCT order_id) AS DECIMAL) AS frequency,
    CAST(SUM(payment_value) AS DECIMAL) AS monetary
  FROM final_table
  GROUP BY customer_unique_id
)
SELECT
  ft.*,
  rf.recency,
  rf.frequency,
  rf.monetary
FROM final_table ft
JOIN rfm rf
ON ft.customer_unique_id = rf.customer_unique_id;
```



Valeurs manquantes - Python

Après avoir identifié les valeurs aberrantes, je me concentre sur le traitement des données manquantes.

- product_id
 - seller_id
 - product_category_name
 - product_category_name_english
 - seller_city
 - seller_state
 - review_id
- 
- Absent

- seller_zip_code_prefix
 - price
 - freight_value
 - payments_installments
- 
- 0

• payment_type → not_defined

• payment_values → price + freight_values

Pour la variable review_score:

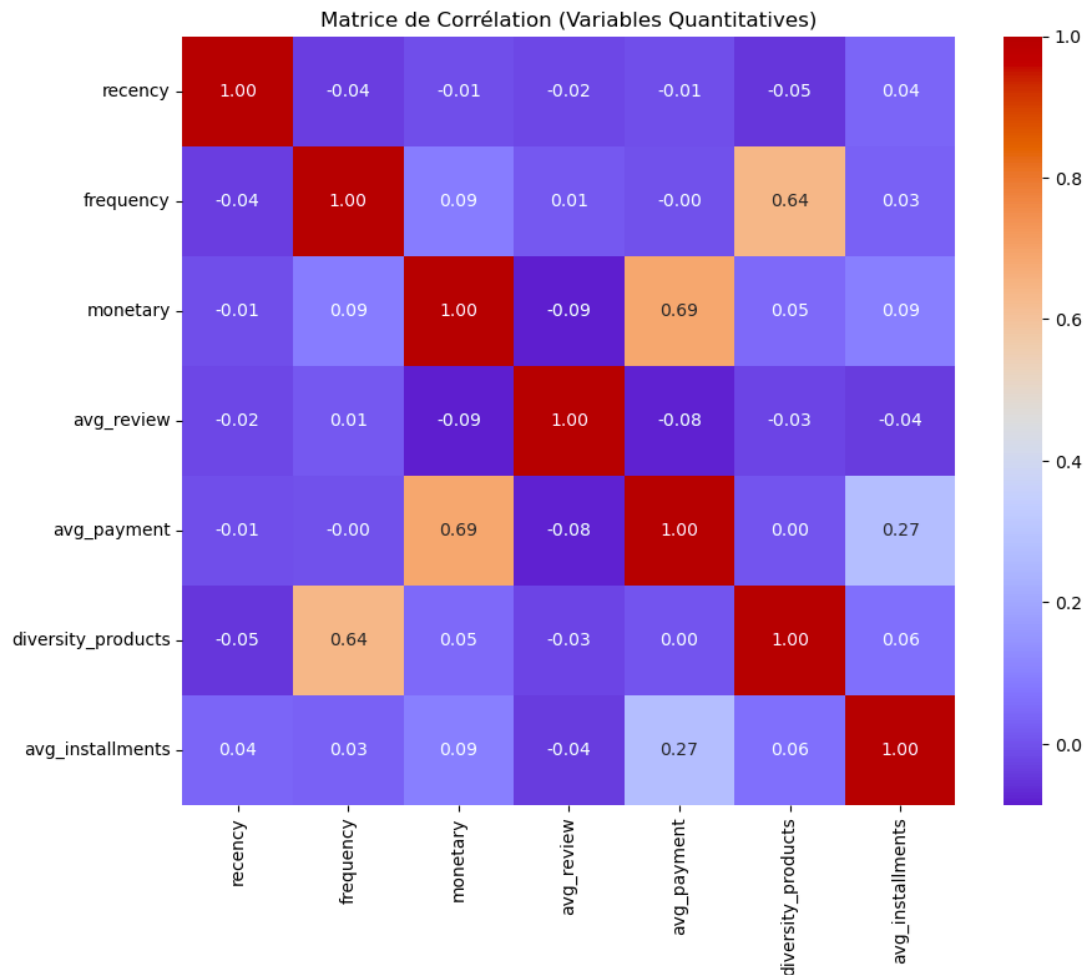
1. Création d'une variable binaire « *has_review* » → 0 si le client n'a pas laissé d'avis
→ 1 s'il en a laissé un.
2. Création de la variable avg_review → la satisfaction moyenne par client uniquement pour ceux ayant fourni un avis.
3. Valeurs manquantes de avg_review → médiane afin de ne pas perturber la distribution.

Cette méthode représente fidèlement la réalité des avis tout en préservant un maximum d'informations.

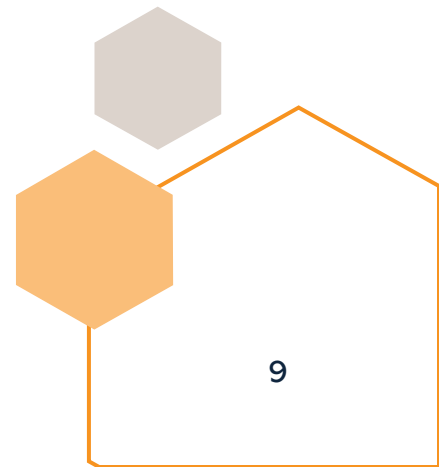
Création de variables

Has_review	Avg_review	Avg_payments	Avg_installments	Diversity_products	Top_category
<u>Variable binaire:</u> 0 = le client n'a pas laissé d'avis 1 = le client a laissé un avis	Valeur moyenne de la note laissée par client	Valeur moyenne dépensée par client	Nombre moyen d'échelonnage de paiement par clients	Nombre de catégories distinctes achetées par un client	Nom de la catégorie produit préférée par client
Informe sur l'engagement du client	Informe sur la satisfaction générale du client	Informe sur les dépenses moyennes du client	Informe sur la capacité de paiement du client	Informe si le client est diversifié ou spécifique dans ses achats	Informe sur les préférences du client

Corrélation variable quantitatives



Cette matrice nous aide à visualiser les coefficients de corrélation entre les features quantitatives. Nous observons que la variable « *avg_payments* » présente une corrélation positive avec « *monetary* », tandis que « *diversity_products* » est corrélée avec « *frequency* ». Ces variables pourraient potentiellement fournir des informations similaires.



Modèle Kmeans

Clustering partitionnel : regroupe les données en k clusters distincts en minimisant la distance intra-clusters.

Objectif : minimiser la distance intra-cluster et maximiser la séparation entre clusters.

Modèle de base : RFM

Ajout de chaque variable au modèle de base

Choix du nombre optimal de clusters :

Méthode du coude : identifie le point d'inflexion où ajouter un cluster n'améliore plus significativement l'homogénéité intra clusters.

Silhouette Score : évalue la qualité de séparation des clusters, est compris entre -1 et 1 (1 = parfaitement séparé). Il fournit une mesure locale, évaluant la qualité du clustering pour chaque point individuellement.

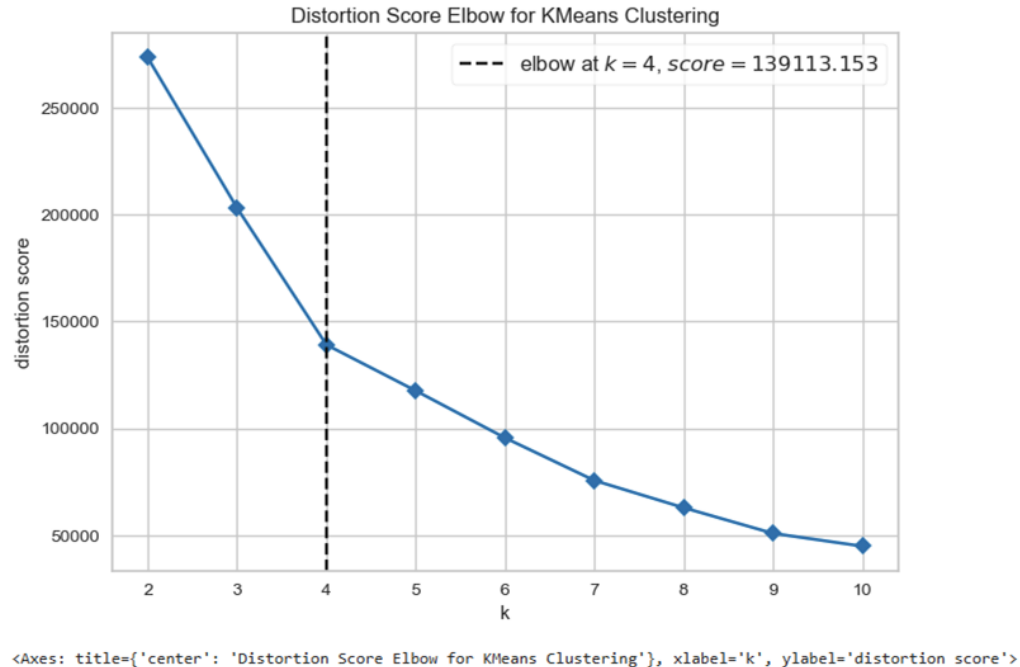
Évaluation des clusters :

Indice Calinski-Harabasz (CH) : compare la compacité intra-cluster et la séparation inter-cluster en faisant le rapport entre la dispersion inter clusters et intra clusters (plus il est élevé plus les clusters sont bien séparés et compacts).

Indice Davies-Bouldin (DB) : mesure la similarité entre clusters (plus il est bas, plus les clusters sont bien distincts).

CH et DB offrent une vue globale sur la qualité du clustering.

Modèle de base: RFM



Silhouette score pour 4 clusters: 0,53
Silhouette score pour 5 clusters: 0,54
Silhouette score pour 6 clusters: 0,54

Métriques:
CH score: 60721
DB score; 0,61

Moyennes en échelle d'origine			
Cluster	Recency	Frequency	Monetary
0	177,75	1,00	314,95
1	178,14	7,29	1245,88
2	438,32	1,00	309,62
3	242,35	1,74	39038,20
4	275,83	2,14	755,42

Distinction des groupes de clients :

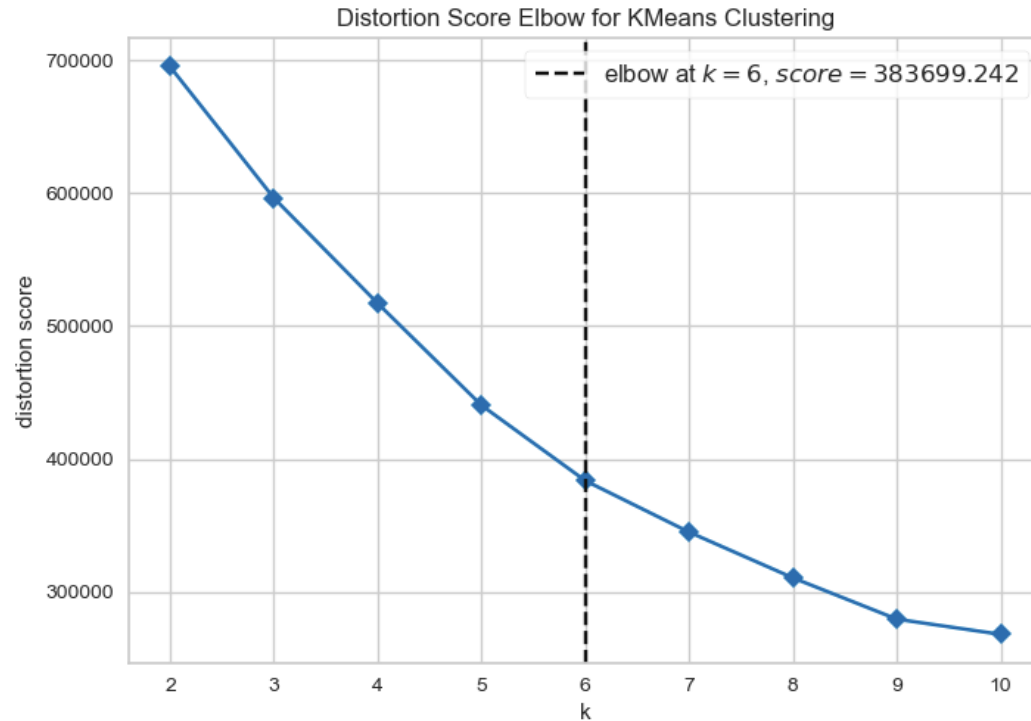
- Cluster 0 : Nouveaux clients ayant passé une seule commande avec des dépenses faibles.
- Cluster 1 : Clients réguliers qui effectuent des achats fréquents et dépensent des montants élevés.
- Cluster 2 : Clients inactifs ayant réalisé une unique commande avec des dépenses très limitées.
- Cluster 3 : Clients avec des dépenses très élevées.
- Cluster 4 : Clients intermédiaires qui passent des commandes régulièrement et dépensent des sommes raisonnables.

Synthèse des variantes

Variante	Nombre clusters	Silhouette score	Indice CH	Indice DB
Avg_review	5	0,45	49815	0,74
Has-review	5	0,44	49522	0,75
Top_category 1	5	0,26	30793	1,09
Top_category 2	5	0,26	30689	1,09
Avg_payment	5	0,38	34225	0,87
Avg_installments	6	0,35	35536	0,88
Diversity_products	5	0,43	46467	0,77

- « Top_category » impacte négativement les métriques: la distinction entre les clusters devient plus complexe. Les diverses catégories de cette variable introduisent un bruit que le modèle peine à gérer efficacement.
- Les variables monétaires (« avg_payment » et « avg_installment ») affectent également les métriques, seule la variable « avg_installment » fournit une information complémentaire au profil des clients.
- Les variables « avg_review », « has_review » et « diversity_products » apportent des informations essentielles à la segmentation des clients et conservent des scores satisfaisants.

Modèle avec toutes les variables



Silhouette score pour 5 clusters: 0,24
Silhouette score pour 6 clusters: 0,20

Métriques pour 5 Clusters

CH score: 21353
DB score: 1,29

```
Cluster
0      top_category_bed_bath_table
1      top_category_bed_bath_table
2  top_category_computers_accessories
3      top_category_bed_bath_table
4      top_category_health_beauty
dtype: object
```

Moyennes en échelle d'origine:					
	recency	frequency	monetary	avg_review	avg_payment
Cluster					
0	308.345751	1.042996	719.820560	4.209235	374.751488
1	293.086181	1.032903	322.617524	1.802455	149.068350
2	271.216718	1.445820	23747.874768	2.523220	2411.899752
3	258.894659	2.058261	645.432233	3.868324	168.442635
4	283.523435	1.027975	186.151628	4.746651	119.375879

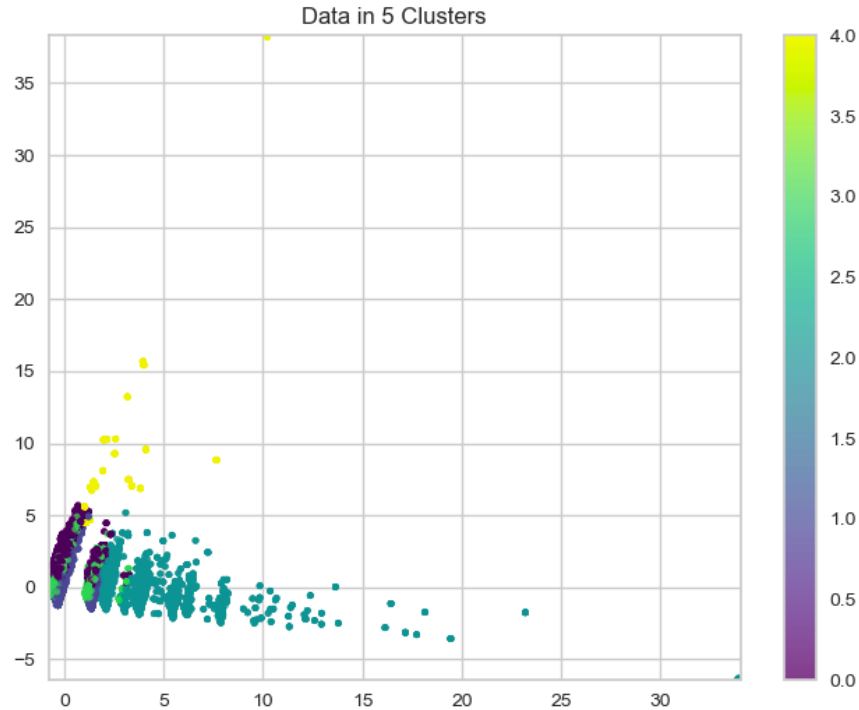
	diversity_products	avg_installments	has_review
Cluster			
0	1.007138	8.007584	0.989070
1	1.000253	2.230511	1.000000
2	1.061920	4.312693	0.934985
3	2.109166	3.422686	0.991761
4	1.000000	1.836961	0.989714

Identification des segments de clients :

- [Cluster 0](#) : inactivité et haut niveau d'échelonnement de paiement
- [Cluster 1](#) : taux d'insatisfaction élevé et très engagés pour laisser un avis
- [Cluster 2](#) : dépenses très élevées, peu engagés pour laisser un avis
- [Cluster 3](#) : clients récents, fréquence d'achat élevée, achat de produits variés.
- [Cluster 4](#) : faibles dépenses et grande satisfaction.

Modèle final

Variables RFM, avg_review, has_review, diversity_products et avg_installments



Silhouette Score for 5 clusters: 0.31

Silhouette Score for 6 clusters: 0.34

Métriques pour k=5:

Indice CH : 30915

Indice DB : 1.04

Moyennes en échelle d'origine:

	recency	frequency	monetary	avg_review	diversity_products
Cluster					
0	291.405708	1.035857	463.301200	1.543245	1.000000
1	181.988894	1.027105	246.763789	4.652231	1.000000
2	257.432386	2.038245	709.553601	3.847308	2.107952
3	453.278055	1.035780	285.978277	4.619034	1.000168
4	269.847328	1.492366	26987.416489	2.488550	1.038168

	avg_installments	has_review
Cluster		
0	3.218297	1.000000
1	2.416100	0.989939
2	3.588685	0.991918
3	3.437917	0.989569
4	4.576336	0.923664

Identification des segments de clients :

- Cluster 0 : insatisfaits, très engagés pour donner un avis.
- Cluster 1 : récents, très satisfaits, mais peu dépensiers.
- Cluster 2 : achètent souvent et divers produits.
- Cluster 3 : inactifs, dépensent peu mais sont satisfaits.
- Cluster 4 : grandes dépenses avec des options de paiement, peu engagés pour laisser un avis.

Modèle DBSCAN

DBSCAN (Clustering basé sur la densité) :

- Regroupe les points denses (concentration de points) et identifie les outliers (point isolé, pas assez de point proche dans son rayon).
- Ne nécessite pas de nombre de clusters prédéfini.
- Consomme beaucoup de mémoire → testé sur un sous-échantillon représentatif.

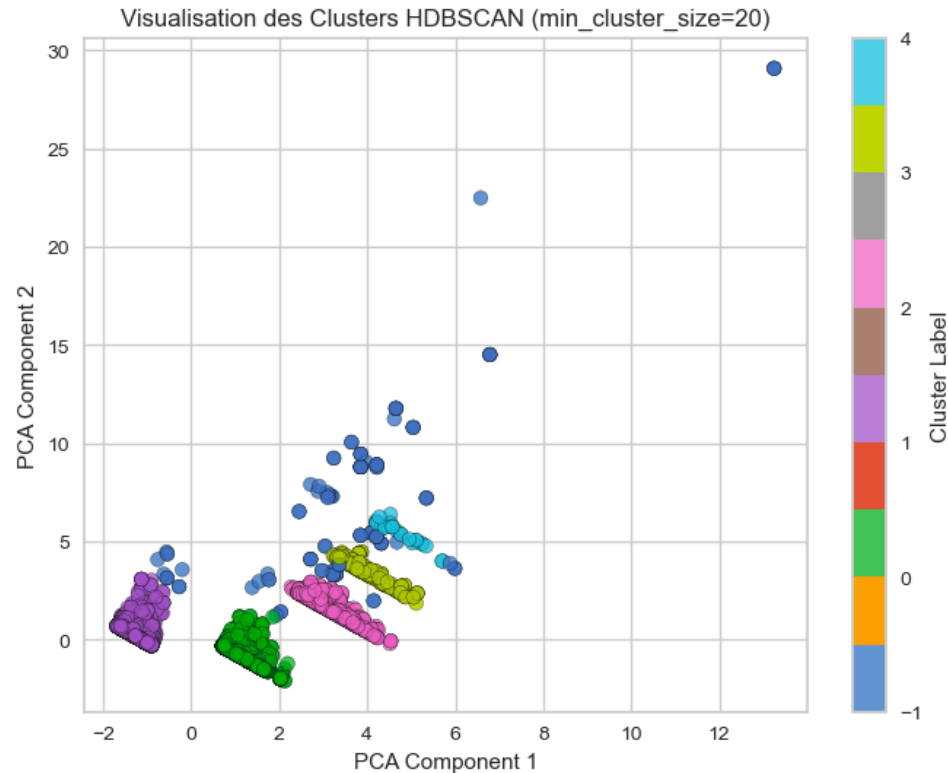
HDBSCAN :

- Version optimisée de DBSCAN (moins gourmand en mémoire).
- Non présenté dans les résultats.

Optimisation des paramètres :

- ϵ (epsilon) : définit le rayon autour d'un point.
 - Utilisation de la courbe des k-plus proches voisins (KNN) pour trouver le point de coude optimal.
- min_samples : nombre minimal de voisins pour qu'un point soit central.
 - Petite base : $\approx 2 \times n_{\text{features}}$.
 - Grande base : $\approx \log(n_{\text{samples}})$ (≈ 11 dans notre cas).
- Évaluation : mêmes métriques que précédemment (Silhouette Score, Indice CH et DB).

Modèle de base: RFM



Moyennes en échelle d'origine			
Cluster	Recency	Frequency	Monetary
0	439,19	1,00	287,05
1	178,73	1,00	301,63
2	281,16	2,00	530,68
3	281,04	3,00	781,61
4	224,42	4,00	932,69

Silhouette Score (sans outliers): 0.72

Indice DB (sans outliers): 0.64

Indice CH(sans outliers): 32372

Silhouette score plus élevé qu'avec le modèle Kmeans mais indice CH beaucoup plus faible, les clusters seraient moins bien séparés globalement.

Identification des segments de clients :

- [Cluster 0](#) : Clients inactifs et dépensant peu
- [Cluster 1](#) : Clients récents
- [Cluster 2](#) : Clusters difficiles à différencier
- [Cluster 3](#) : Clusters difficiles à différencier
- [Cluster 4](#) : Clients fréquents qui dépensent beaucoup

Synthèse des variantes

Variante	Nombre clusters	Silhouette score	Indice CH	Indice DB
Avg_review	4	0,88	848	0,07
Has-review	4	0,88	847	0,07
Top_category 2	4	0,85	653	0,09
Avg_payment	4	0,87	1094	0,08
Avg_installments	6	0,78	454	0,14
Diversity_products	5	0,58	2855	0,90
All variables	5	0,82	518	0,11
Final	4	0,51	2364	1,07

- DBSCAN présente de meilleurs scores de silhouette et DB, mais l'indice CH reste très faible par rapport à K-means, indiquant une séparation insuffisante des clusters.
- La segmentation des profils clients est complexe et peu cohérente avec une répartition déséquilibrée des observations : la majorité des clusters sont quasi vides tandis qu'un seul regroupe une grande partie des données. Ainsi, l'analyse métier n'est pas concluante avec ce modèle, de la même manière qu'avec HDBSCAN, qui présente des problématiques similaires en termes de répartition et d'interprétabilité des résultats

Contrat de maintenance

Méthode utilisée:

1. Découpage de la période analysée

- Période de référence : du 04/09/2016 au 31/12/2017.
- Test du contrat de maintenance sur différents pas: trimestriel, bimestriel (tous les deux mois), mensuel et toutes les deux semaines.

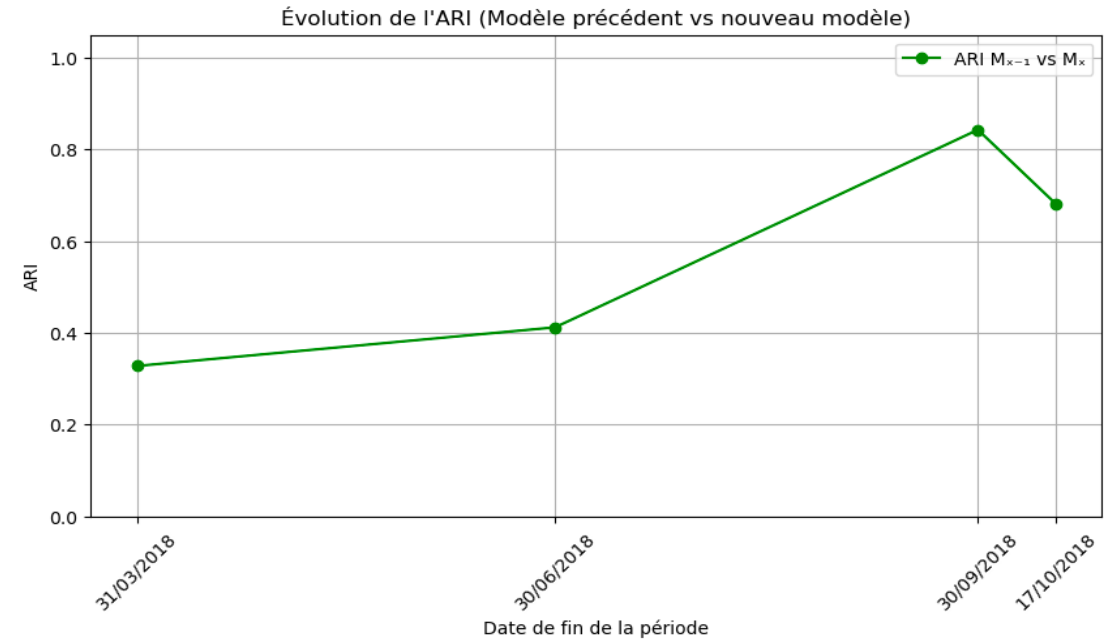
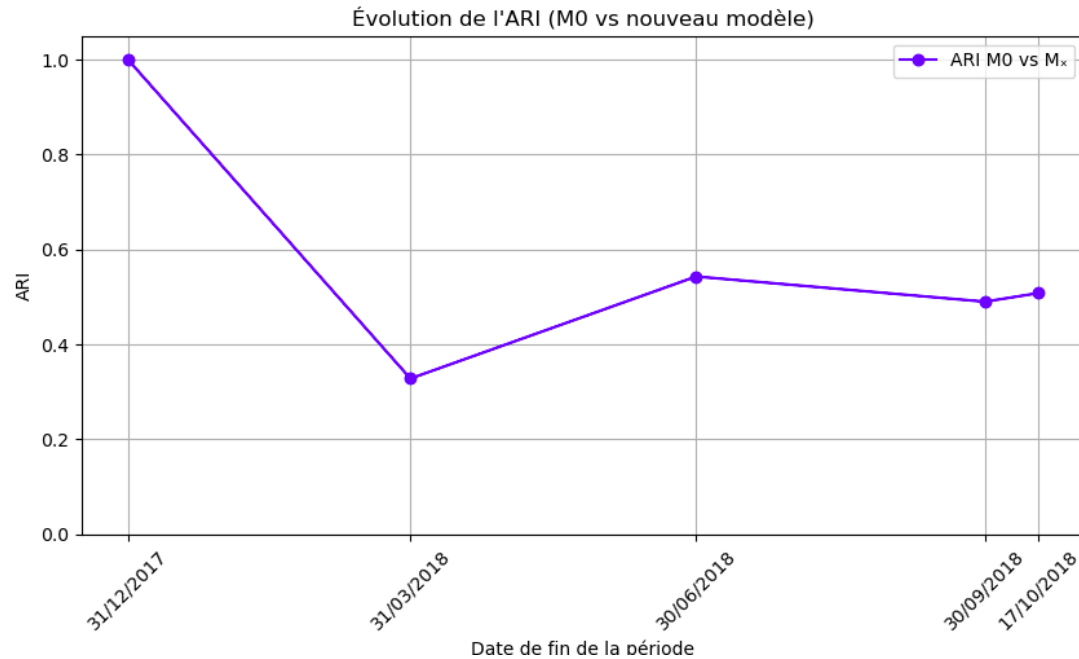
2. **Calibration du modèle de référence (M0):** Le modèle sélectionné est recalculé sur la période de référence.

3. **Application du modèle sur la période n+1:** La transformation (issue du modèle M0) est appliquée à la période n+1 sans recalibrage, afin d'obtenir les clusters correspondant au modèle de référence.

4. **Recalibrage:** Les paramètres sont recalibrés sur la période n+1 pour définir de nouveaux clusters.

5. **Comparaison:** L'ARI est calculé entre le modèle de référence (M0) et le modèle recalibré de n+1, ainsi que, lorsque possible, entre le modèle n-1 et le modèle recalibré n+1, afin d'évaluer la stabilité et la cohérence des clusters.

Maintenance trimestrielle

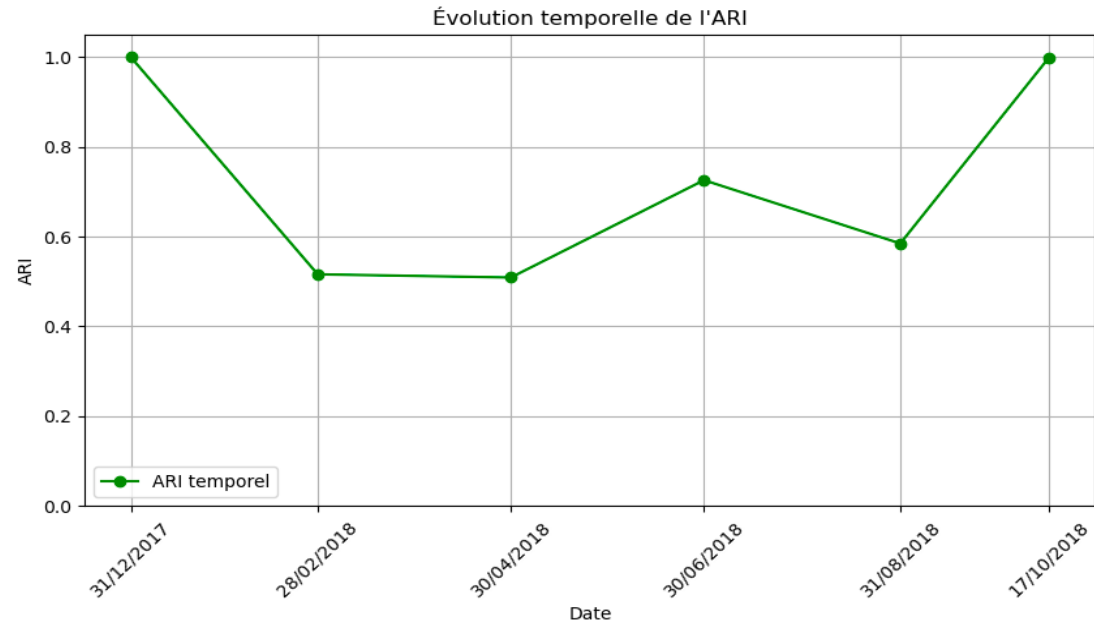
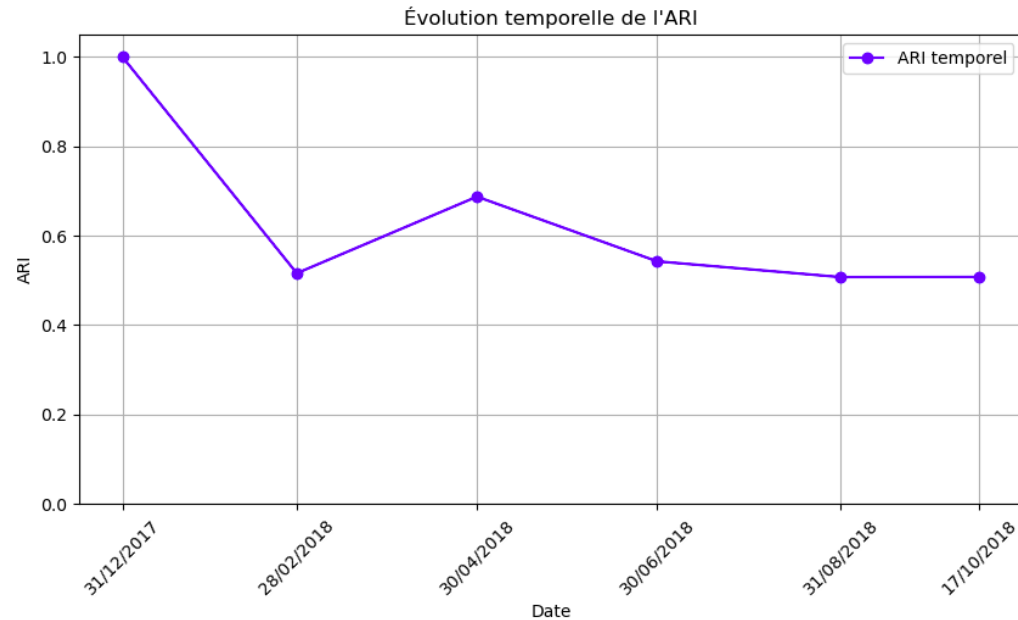


Comparaison avec M0 : Évalue la stabilité à long terme et offre une vue d'ensemble : dans quelle mesure les comportements des clients sont similaires à ceux du modèle de référence ?

Après trois mois, l'ARI descend en dessous de 0,4. Il y a un changement significatif dans le comportement des clients, bien qu'une faible convergence soit observée à la fin juin.

Comparaison avec N-1 : Évalue la stabilité à court terme et fournit un suivi progressif, étape par étape : comment les comportements évoluent-ils d'une période à l'autre ? Les comportements des clients entre juin 2018 et septembre 2018 sont similaires.

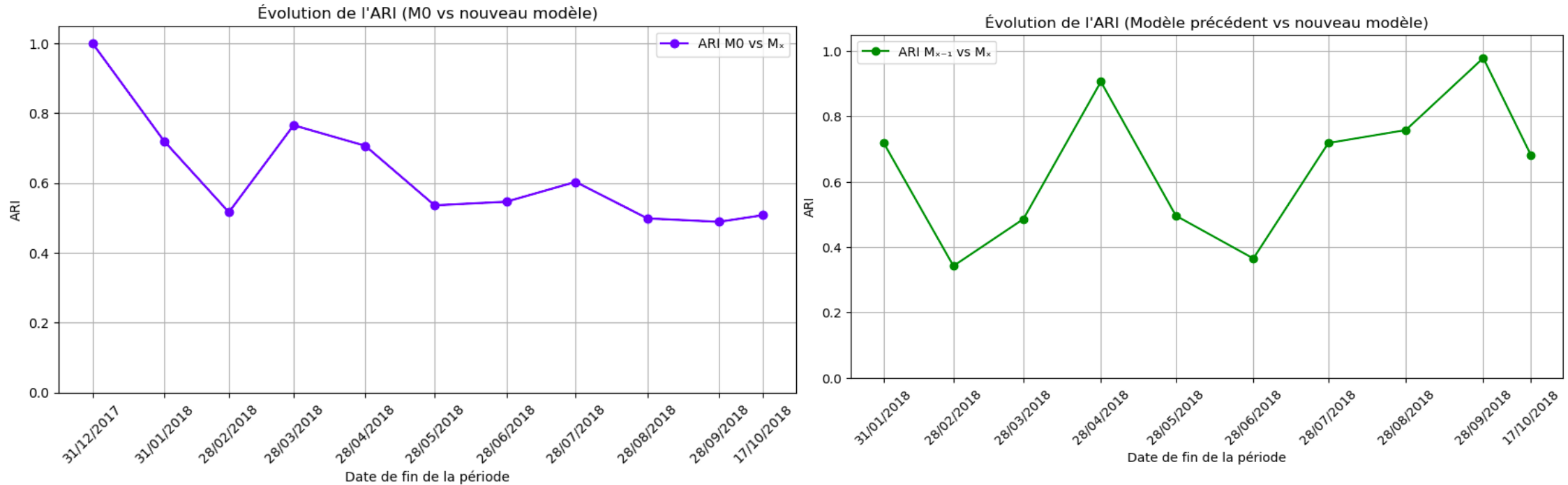
Maintenance bimestrielle



Comparaison avec M0 : L'ARI chute en dessous de 0,6 après deux mois, ce qui indique un changement significatif dans les comportements et la nécessité de revoir la segmentation. Une légère augmentation à la fin avril 2018 suggère que les comportements commencent à s'aligner avec ceux du modèle de référence.

Comparaison avec M-1 : Les comportements semblent ne se rapprocher de ceux de la période antérieure que durant la période de juin. Ils sont identiques entre octobre et septembre 2018.

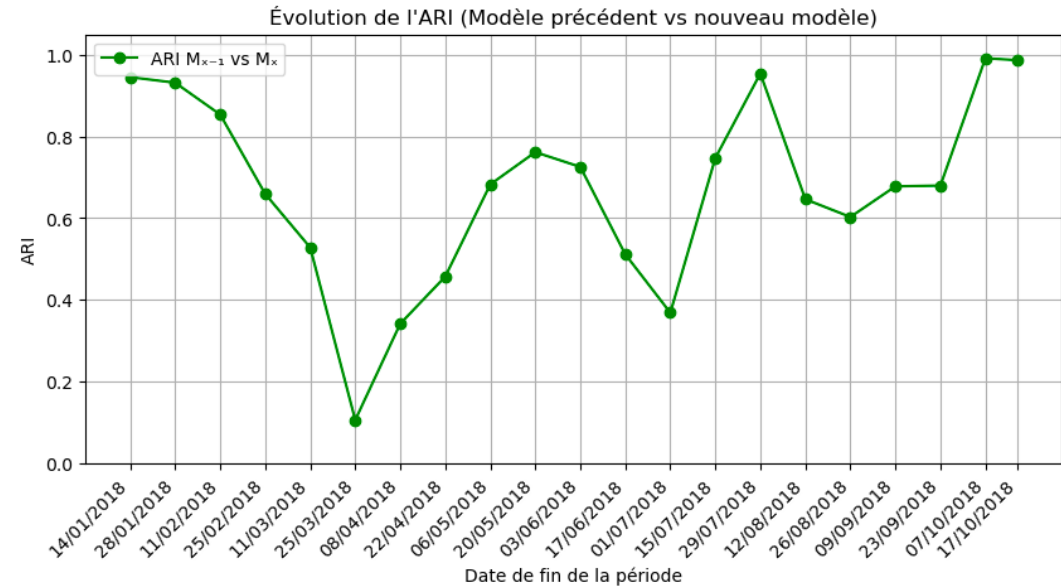
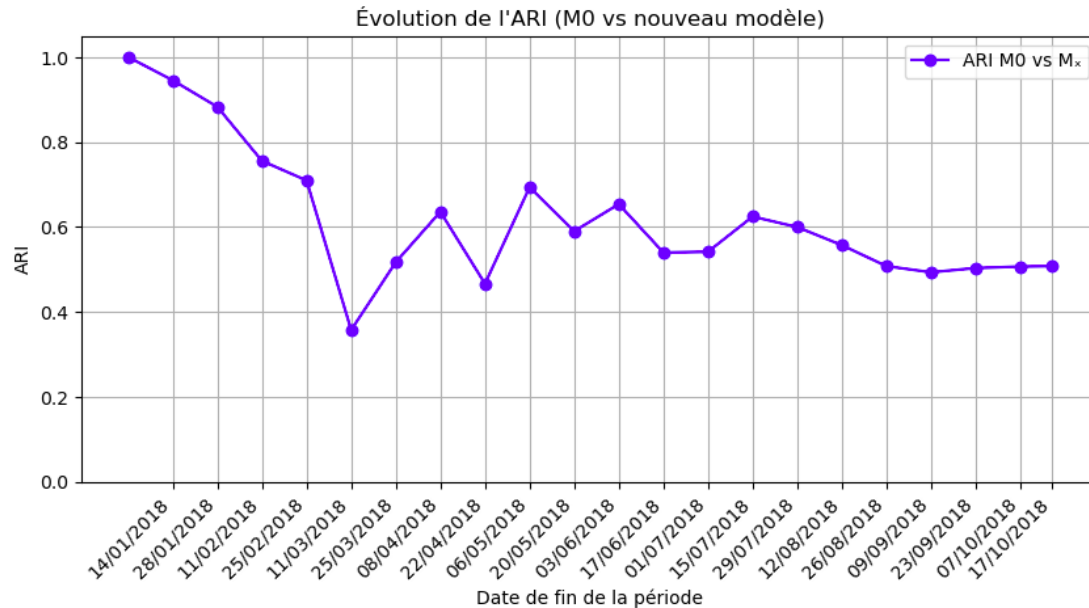
Maintenance mensuelle



Comparaison avec M0 : L'ARI baisse en dessous de 0,8 après un mois, cela suggère encore un changement dans les comportements des clients. Nous retrouvons l'alignement des comportements avec ceux du modèle de référence fin mars. La maintenance mensuelle a permis d'être un peu plus précise mais ne nous permet pas encore d'établir une date exacte de maintenance.

Comparaison avec M-1 : Il y a clairement des oscillations de comportements selon les périodes. Au mois d'avril les comportements sont presque similaires à ceux du mois de mars. De même, les comportements du mois de septembre sont aussi semblables à ceux du mois d'août.

Maintenance 2 semaines



Comparaison avec M0 : L'ARI est inférieur à 0,8 à partir du 11/02/2018, cela suggère que c'est à partir de cette date que les comportements des clients changent notablement avec ceux de la période de référence.

Comparaison avec M-1 : La comparaison avec la période N-1 indique la même analyse. Les comportements sont significativement différents fin février 2018 (par rapport à mi-février 2018). Il y a de fortes variations selon les dates ce qui montrent que les comportements des clients peuvent rester stables (fin juillet, mois d'octobre) où changer drastiquement (fin mars, début juillet).



Conclusion

- L'algorithme **K-means** a été retenu pour effectuer l'analyse client.
- Dimension comportementale **complète**: Le modèle final sélectionné combine les indicateurs de bases RFM, avec des indicateurs de satisfactions, de diversité d'achat ainsi que de comportement de paiement, ce qui donne une vision plus détaillée et actionnable des profils clients d'olist.
- Un **équilibre** a été trouvé entre métriques satisfaisantes et une bonne analyse métier. Les clusters identifiés sont facilement interprétables et intuitifs pour les actions à entreprendre pour le service marketing.
- Le contrat de maintenance suggère une nouvelle analyse **toutes les 6 semaines**.

Merci

Antonine LAROZE-CERVETTI