Segmenter les clients du site de e-commerce Olist

Noura Romari – Parcours Ingénieur Machine Learning depuis le 04/03/22
Soutenance Projet n°4 du 17/08/22

Mentor : Hamza Tajmouati
Evaluateur : Florian Guillet

Sommaire

Introduction

Introduction

- **Contexte**

  - Olist est une entreprise basée à Mercês, au Brésil.
  - Olist propose un panel de services de support à la vente en ligne :
    - Gestion d'entreprise
    - Vente en marketplace
    - Boutique en ligne
    - Optimisation logistique
  - Spécialiste de l'accompagnement des commerces de détails.

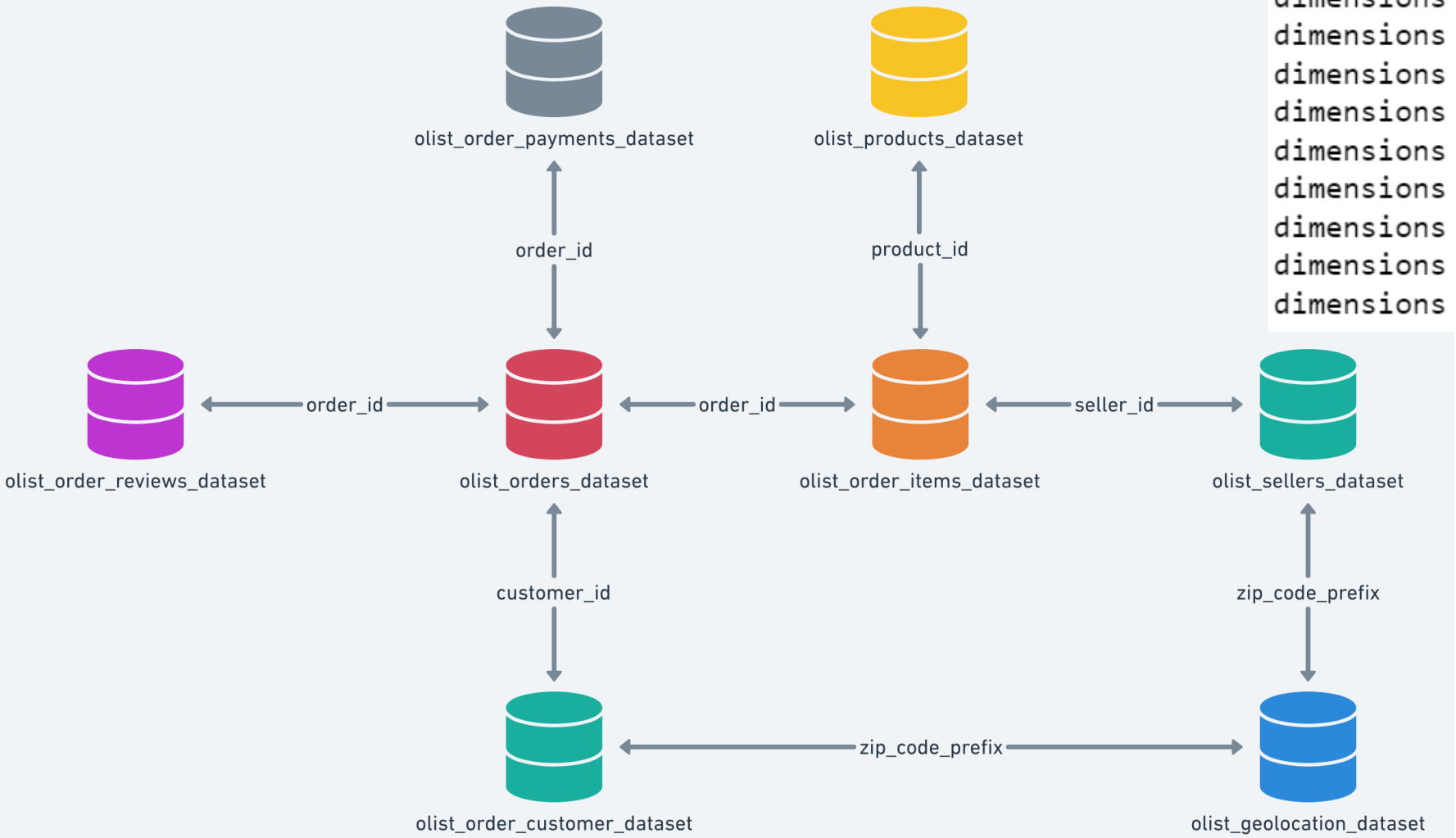
- **Objectifs**

  - Proposer à Olist une segmentation client pertinente et utilisable pour réaliser une communication ciblée.
  - Explorer les données personnelles des clients et analyser leur comportement d'achat.
  - Proposer un contrat de maintenance du modèle de segmentation, basée sur l'analyse de la stabilité des segments au cours du temps.

Introduction

- **Les données**

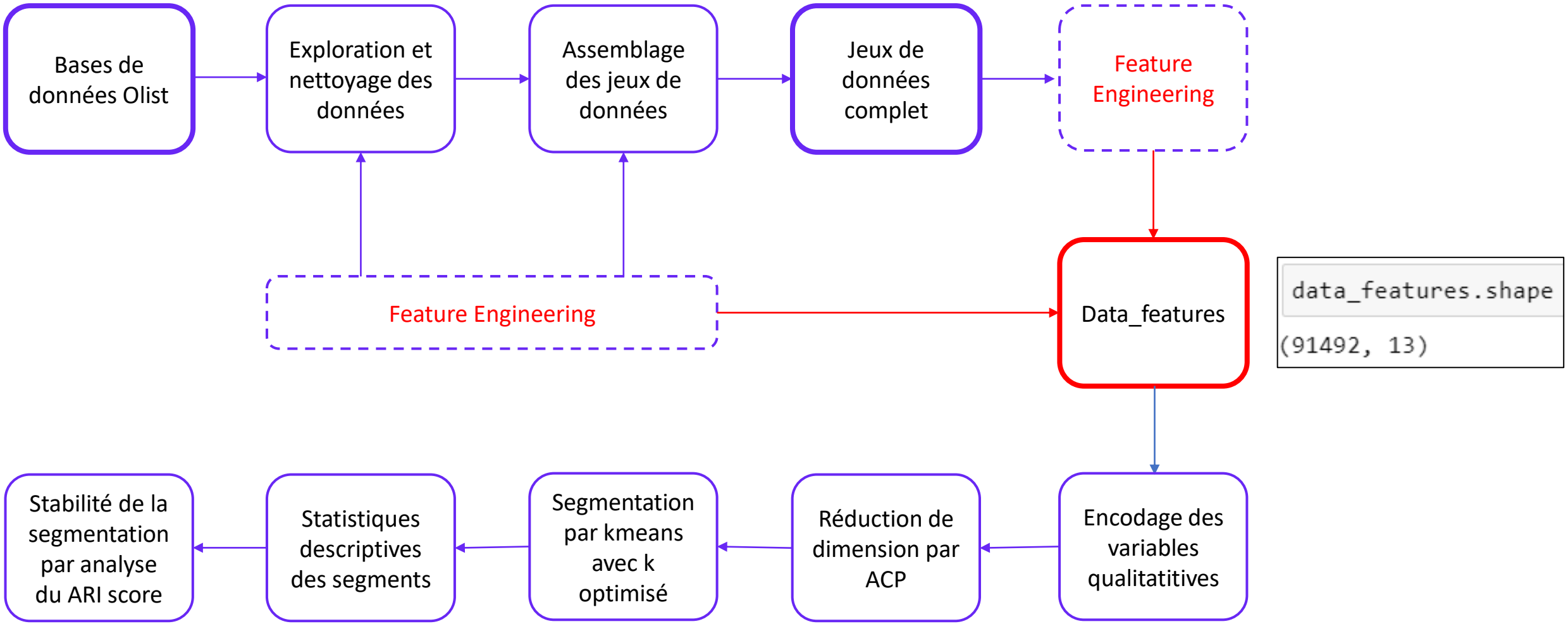
➢ Ensemble de jeux de données de nature variée alimentée depuis 2017




```
dimensions df_customers : (99441, 5)
dimensions df_geolocalisation : (1000163, 5)
dimensions df_order_items : (112650, 7)
dimensions df_order_payments : (103886, 5)
dimensions df_order_reviews : (99224, 7)
dimensions df_orders : (99441, 8)
dimensions df_products : (32951, 9)
dimensions df_sellers : (3095, 4)
dimensions df_translation : (71, 2)
```

Introduction

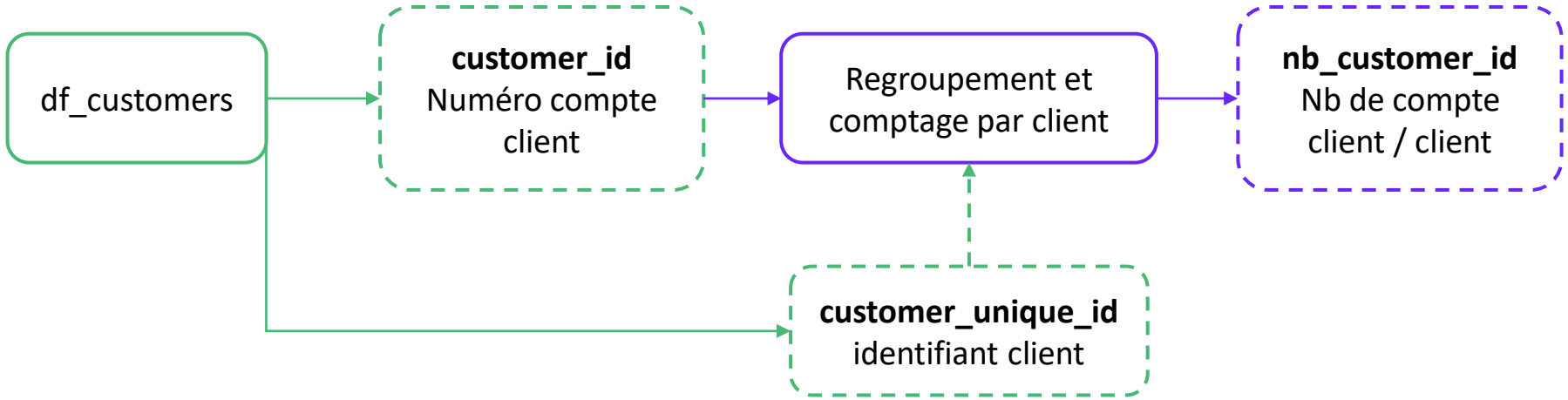
- **Stratégie globale**



Feature Engineering

Feature Engineering

- **Nombre de compte client par client**




```
count(data_features['nb_customer_id'])
```

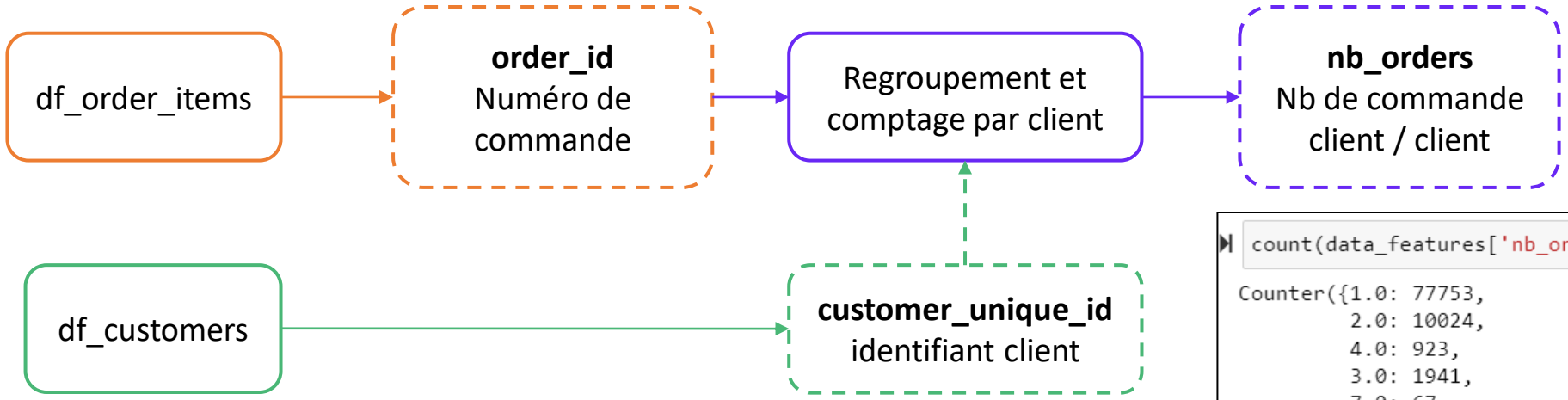
```
Counter({1: 88529, 2: 2713, 3: 201, 4: 30, 6: 6, 7: 3, 5: 8, 9: 1, 17: 1})
```

```
data_features['nb_customer_id'].describe()
```

```
count    91492.000000
mean         1.036167
std          0.218633
min          1.000000
25%          1.000000
50%          1.000000
75%          1.000000
max         17.000000
Name: nb_customer_id, dtype: float64
```

Feature Engineering

- **Nombre de commande par client**




```
data_features['nb_orders'].describe()
```

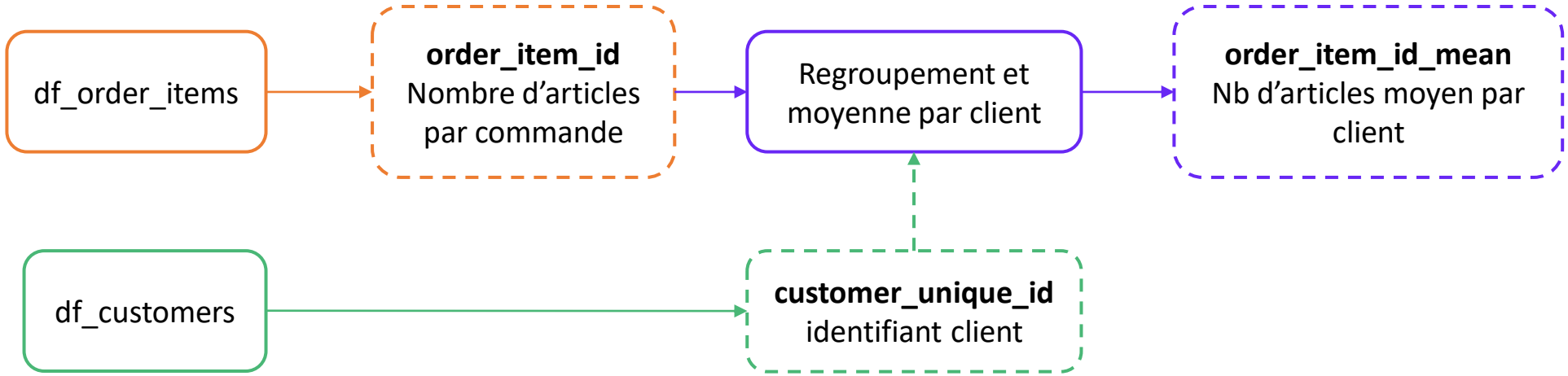
```
count    91492.000000
mean         1.237540
std          0.844299
min          1.000000
25%          1.000000
50%          1.000000
75%          1.000000
max         75.000000
Name: nb_orders, dtype: float64
```

```
count(data_features['nb_orders'])
```

```
Counter({1.0: 77753,
         2.0: 10024,
         4.0: 923,
         3.0: 1941,
         7.0: 67,
         5.0: 325,
         6.0: 284,
         12.0: 25,
         15.0: 6,
         8.0: 45,
         10.0: 24,
         18.0: 1,
         11.0: 17,
         9.0: 24,
         14.0: 9,
         24.0: 7,
         19.0: 1,
         22.0: 1,
         20.0: 3,
         21.0: 3,
         26.0: 1,
         16.0: 1,
         38.0: 1,
         13.0: 4,
         75.0: 1,
         35.0: 1})
```

Feature Engineering

- **Nombre moyen d'articles par commande**

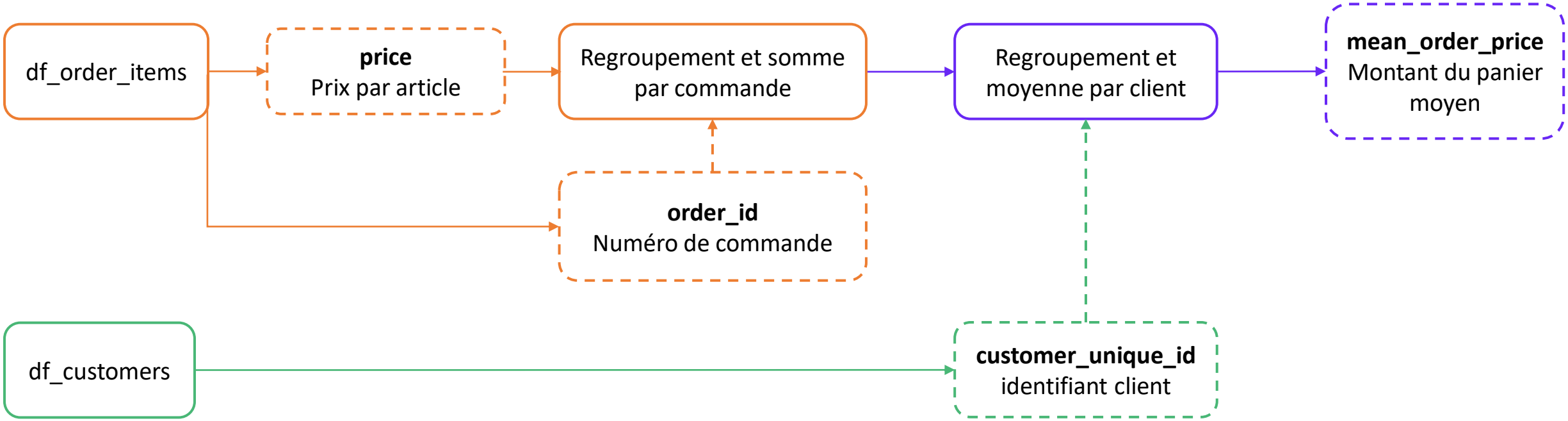



```
data_features['order_item_id_mean'].describe()
```

```
count    91492.000000
mean         1.070659
std          0.265944
min          1.000000
25%          1.000000
50%          1.000000
75%          1.000000
max         11.000000
Name: order_item_id_mean, dtype: float64
```

Feature Engineering

- **Montant du panier moyen**

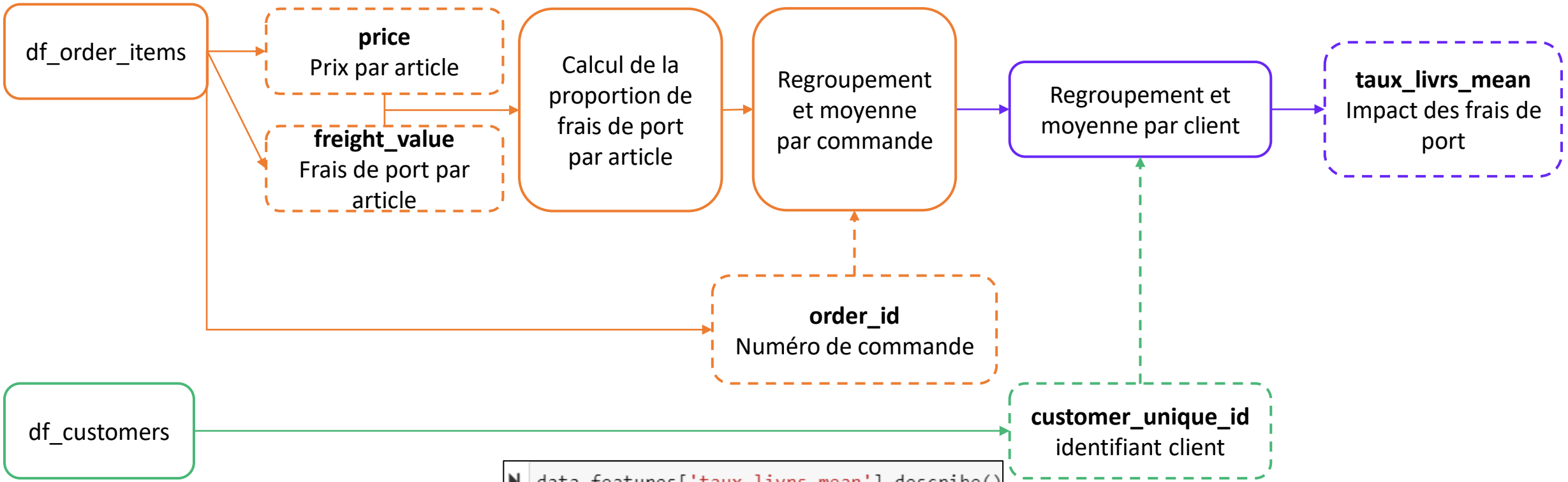



```
data_features['mean_order_price'].describe()
```

```
count    91492.000000
mean       137.520862
std        208.615637
min          0.850000
25%         46.039107
50%         86.900000
75%        149.900000
max      13440.000000
Name: mean_order_price, dtype: float64
```

Feature Engineering

- **Impact des frais de livraison dans l'acte d'achat**

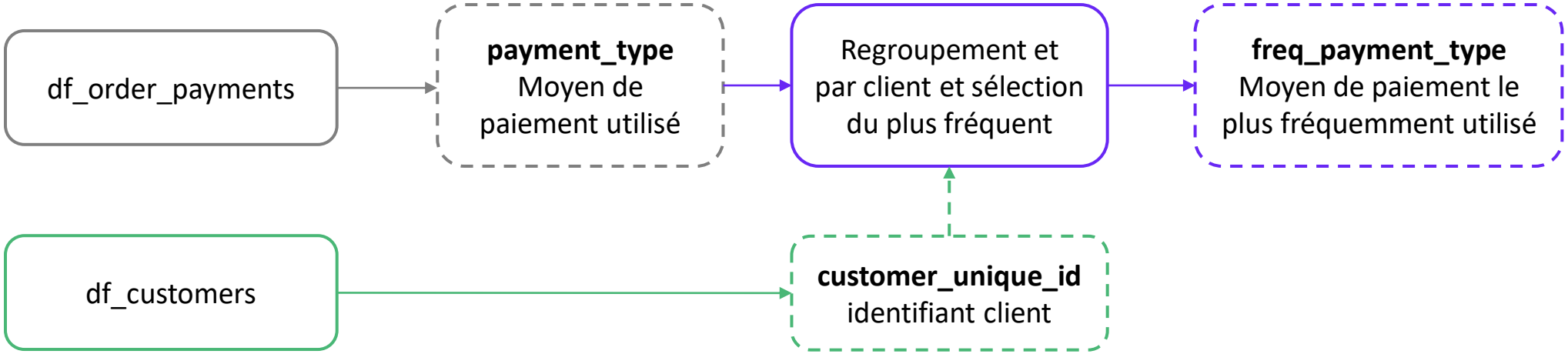



```
data_features['taux_livrs_mean'].describe()
```

```
count    91492.000000
mean        30.890630
std         31.308329
min          0.000000
25%         13.000000
50%         23.000000
75%         38.000000
max       2145.000000
Name: taux_livrs_mean, dtype: float64
```

Feature Engineering

- **Moyen de paiement le plus utilisé**




```
count(data_features['freq_payment_type'])
```

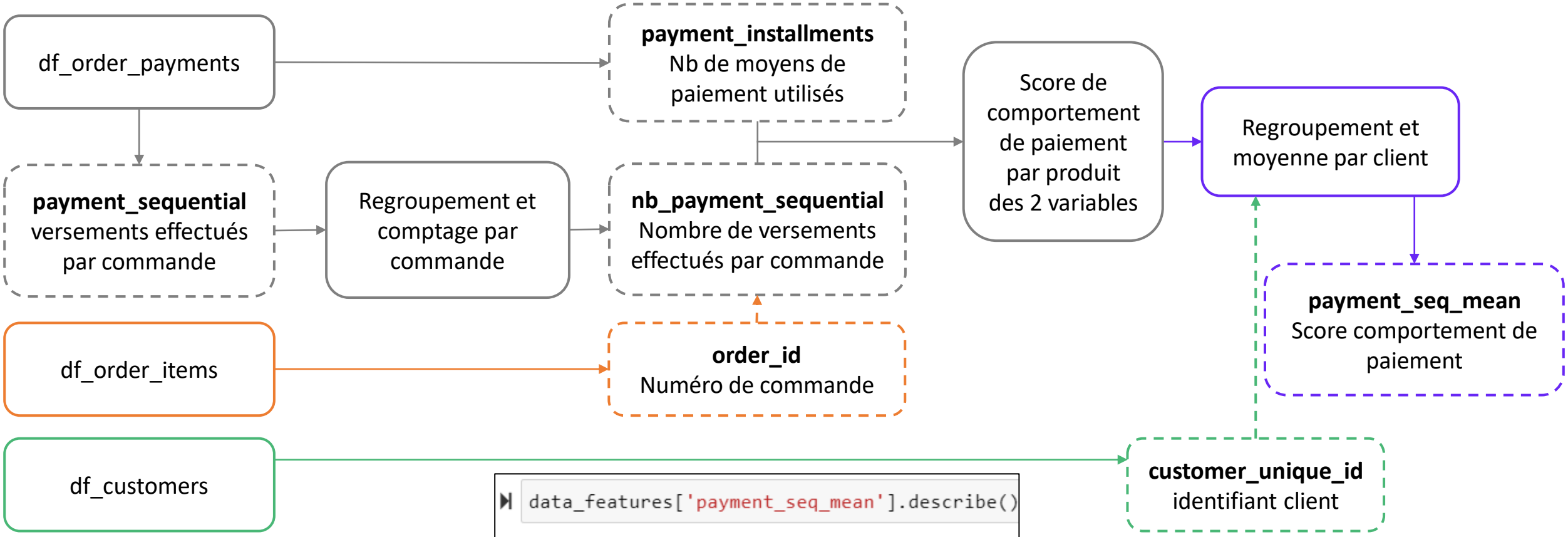
```
Counter({'credit_card': 68433,
         'boleto': 18116,
         'voucher': 3509,
         'debit_card': 1434})
```

```
data_features['freq_payment_type'].describe()
```

```
count            91492
unique               4
top       credit_card
freq             68433
Name: freq_payment_type, dtype: object
```

Feature Engineering

- **Comportement de paiement**

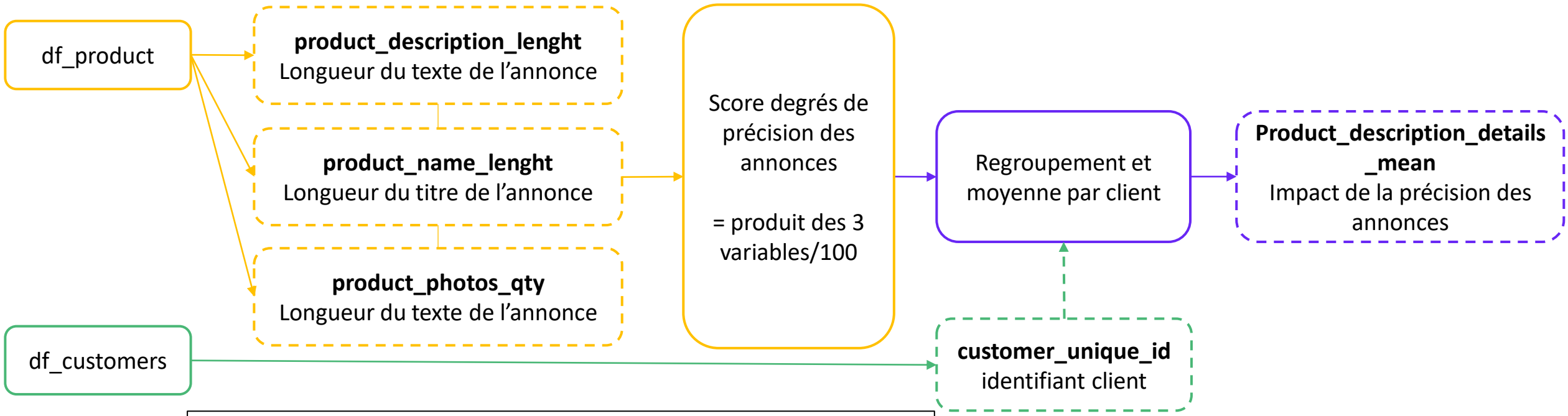



```
data_features['payment_seq_mean'].describe()

count    91492.000000
mean         2.970932
std          2.764153
min          0.000000
25%          1.000000
50%          2.000000
75%          4.000000
max         48.000000
Name: payment_seq_mean, dtype: float64
```

Feature Engineering

- **Impact de la précision des annonces dans l'acte d'achat**

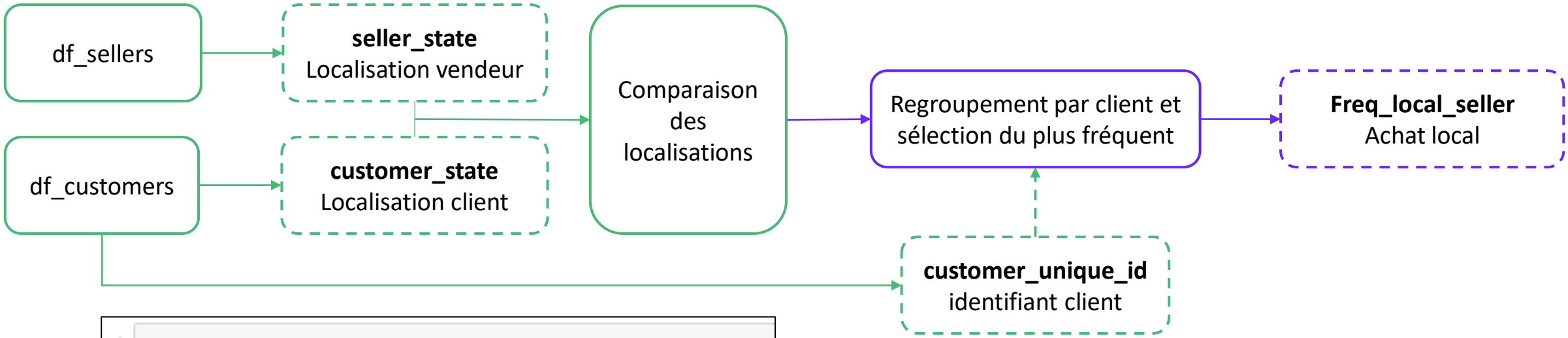



```
data_features['product_description_details_mean'].describe()
```

```
count    91492.000000
mean       985.500793
std       1515.706767
min          1.720000
25%        231.710000
50%        474.800000
75%       1094.400000
max      29128.680000
Name: product_description_details_mean, dtype: float64
```

Feature Engineering

- **Tendance à l'achat en local**




```
count(data_features['freq_local_seller'])
```

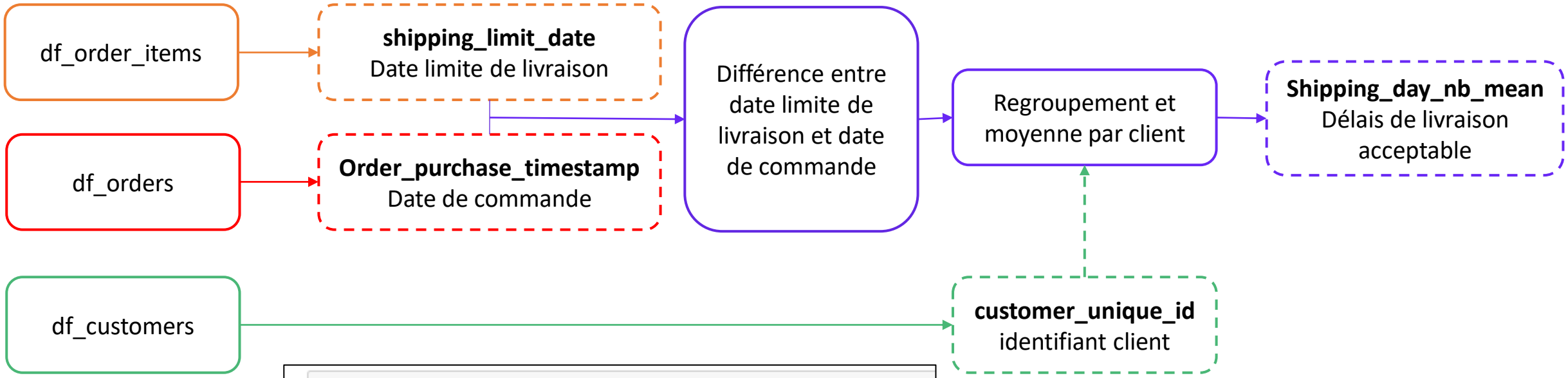
```
Counter({'Yes': 33197, 'No': 58295})
```

```
data_features['freq_local_seller'].describe()
```

```
count     91492
unique        2
top          No
freq      58295
Name: freq_local_seller, dtype: object
```

Feature Engineering

- **Délais de livraison acceptable**

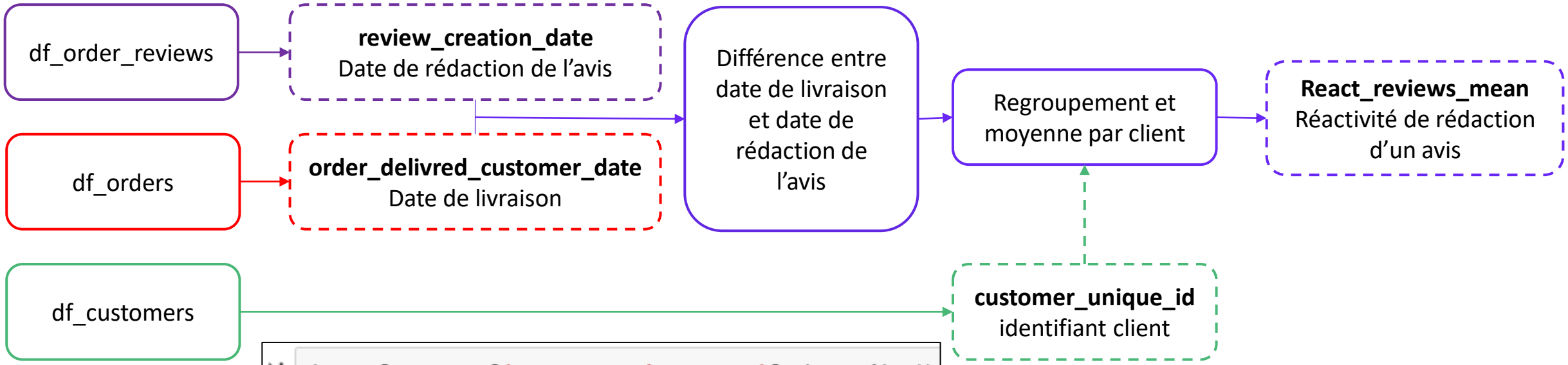



```
data_features['shipping_day_nb_mean'].describe()
```

```
count    91492.000000
mean         6.637521
std          3.658768
min          2.000000
25%          5.000000
50%          6.000000
75%          7.000000
max        529.000000
Name: shipping_day_nb_mean, dtype: float64
```

Feature Engineering

- **Réactivité dans la rédaction des avis**

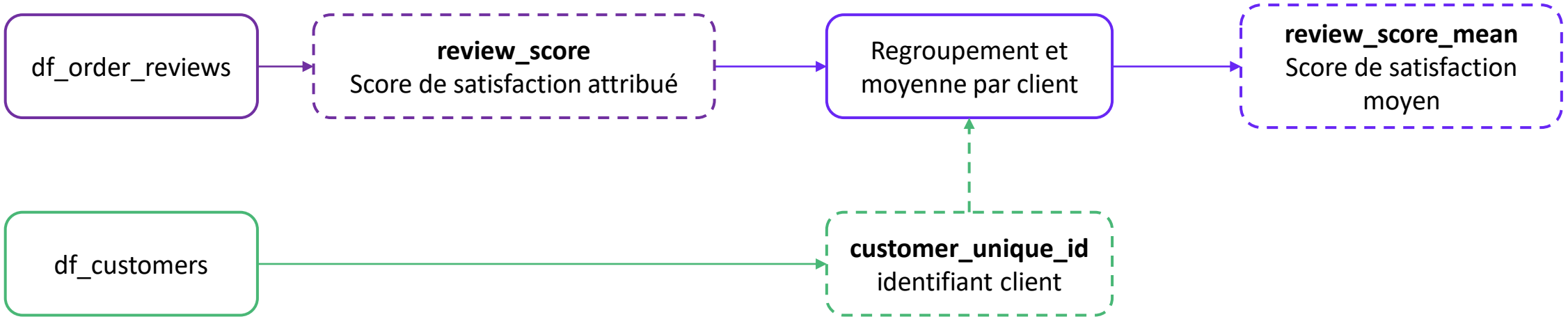



```
data_features['react_review_mean'].describe()
```

```
count    91492.000000
mean         0.453464
std          4.601267
min       -186.000000
25%          1.000000
50%          1.000000
75%          1.000000
max         67.000000
Name: react_review_mean, dtype: float64
```

Feature Engineering

- **Score de satisfaction**

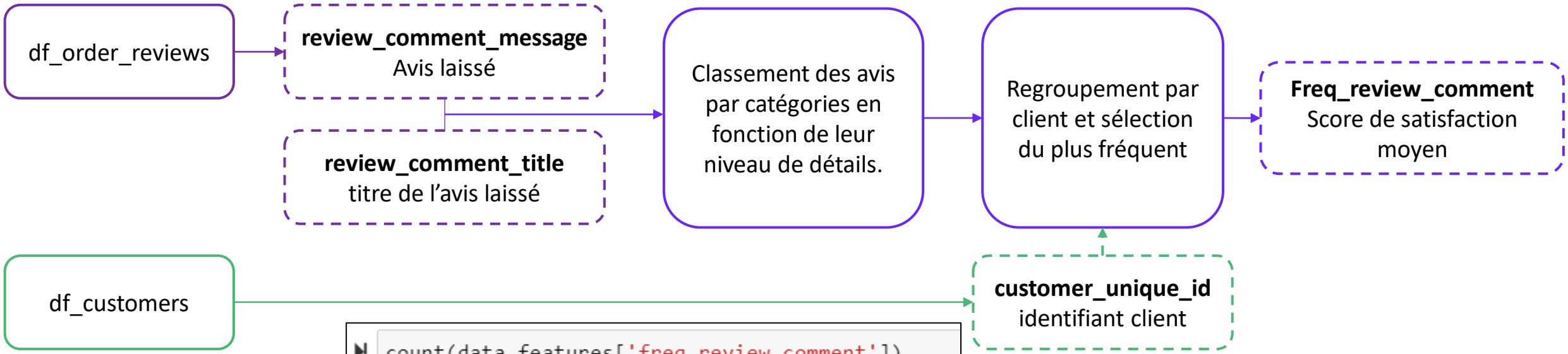



```
data_features['review_score_mean'].describe()
```

```
count    91492.000000
mean         4.154653
std          1.279189
min          1.000000
25%          4.000000
50%          5.000000
75%          5.000000
max          5.000000
Name: review_score_mean, dtype: float64
```

Feature Engineering

- **Comportement lors de la rédaction de l'avis**




```
count(data_features['freq_review_comment'])
```

```
Counter({'Complet message': 9041,
         'No message': 52966,
         'Message': 27803,
         'Short message': 1682})
```

```
data_features['freq_review_comment'].describe()
```

```
count          91492
unique             4
top       No message
freq           52966
Name: freq_review_comment, dtype: object
```

Stratégie de segmentation

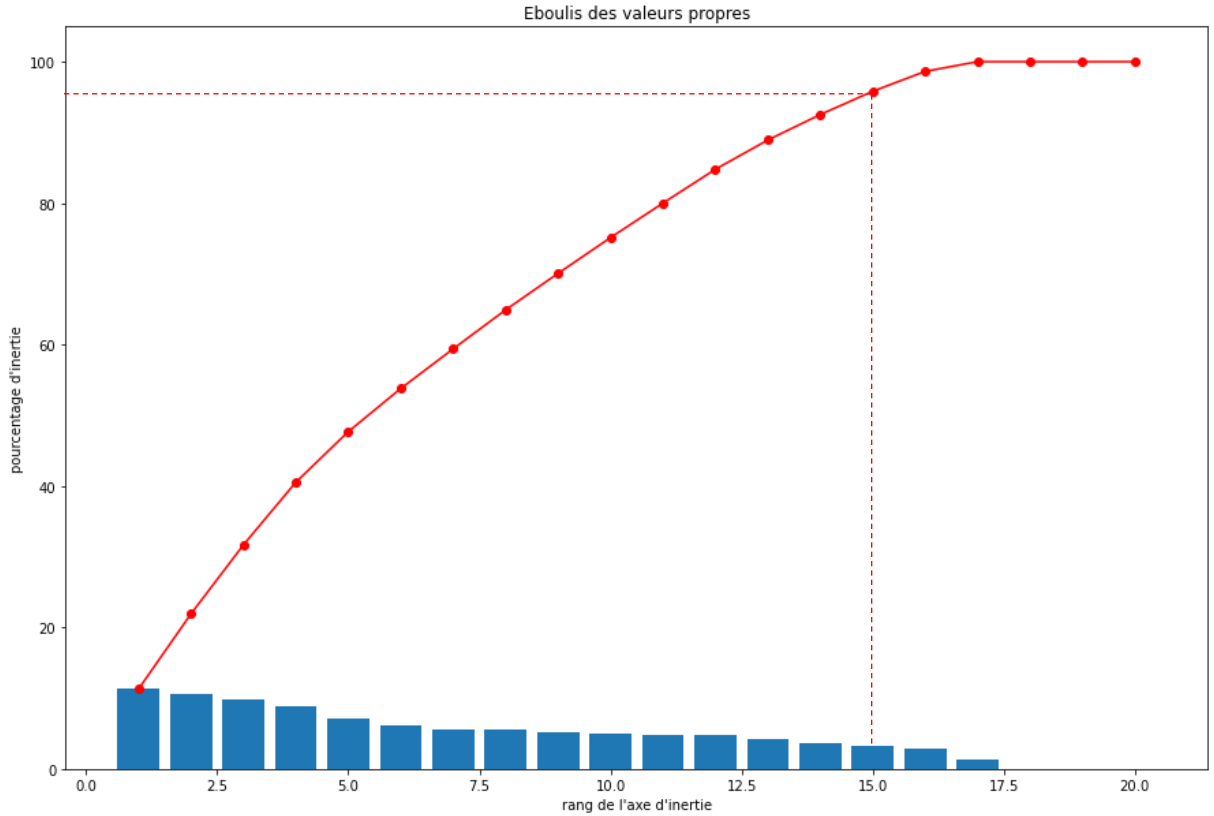
Stratégie de segmentation

- **Encodage des variables qualitatives**

```
data_features.shape
(91492, 13)
```

```
data_all.shape
: (91492, 20)
```

- **Réduction de dimension par ACP**



Je retiens les 15 premières composantes principales qui expliquent 95% de la variabilité.

Stratégie de segmentation

- **Choix du modèle de segmentation**

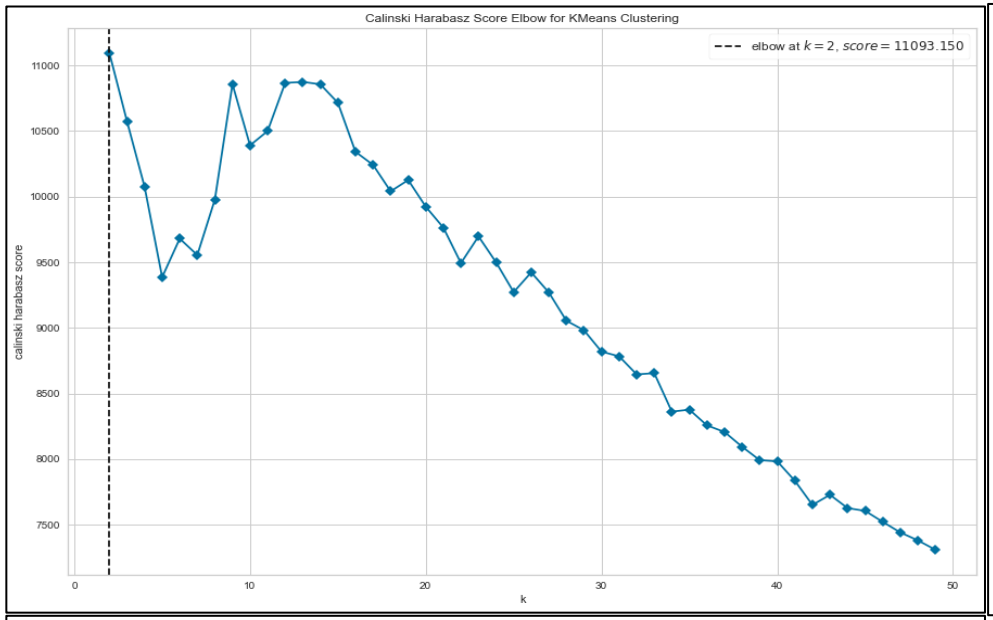
  - Segmentation par kmeans
  - Algorithme de clustering non supervisé
  - Segmentation par minimisation des distances euclidiennes entre individus et centroïdes
  - Paramètre k = nombre de clusters

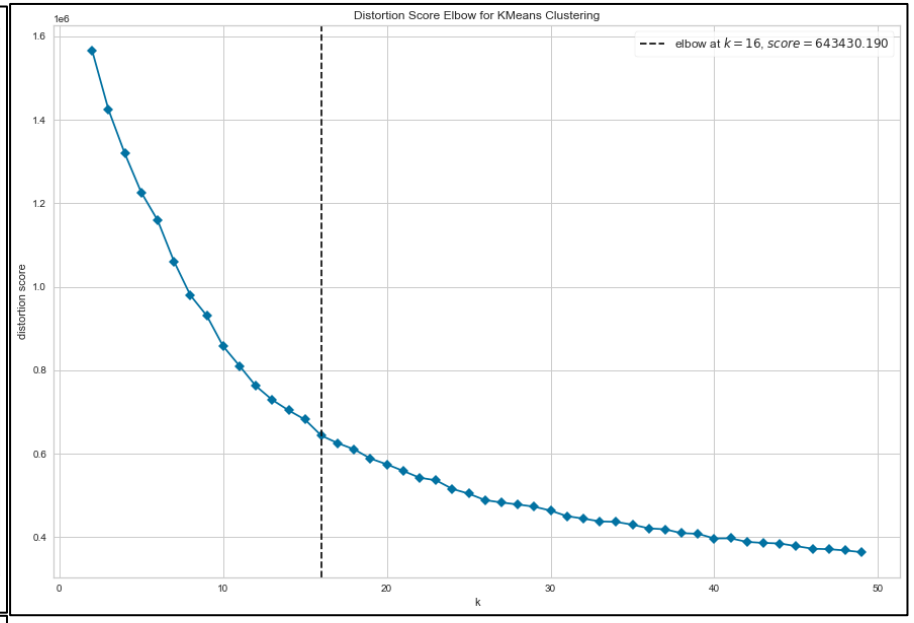
- **Optimisation de k Par Elbow Method**

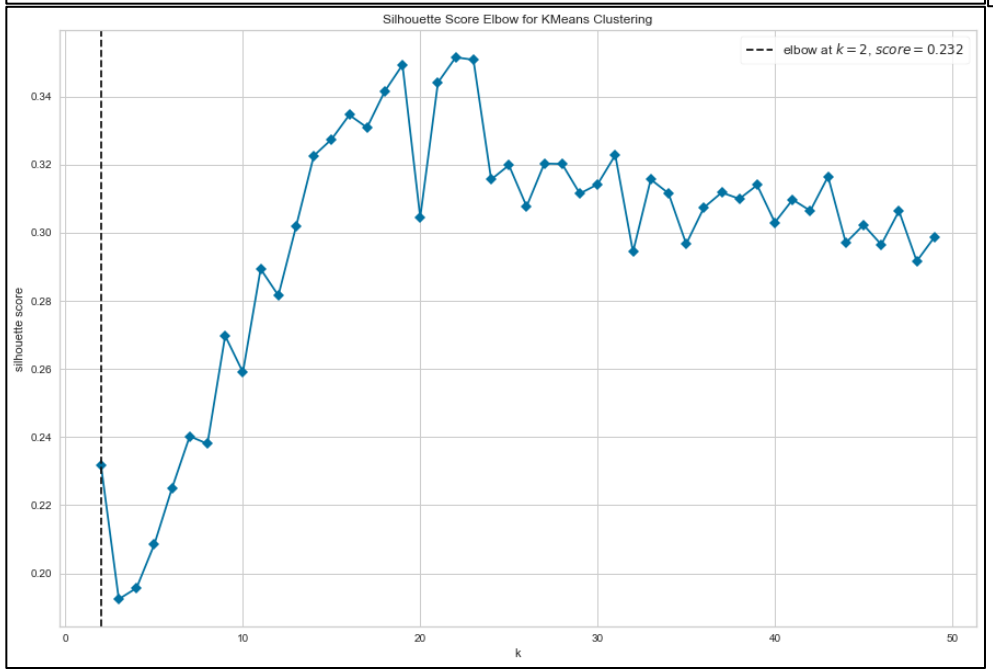
  - Suivi de la variabilité sur une plage définie de valeurs de k
  - 3 métriques de scoring
    - Distortion -> somme des distances au carré entre chaque point et son centroïde
    - calinski_harabasz -> ratio moyen entre la dispersion intra-cluster et inter-cluster
    - Silhouette -> ratio moyen entre la distance intra-cluster et la distance du centroïde le plus proche

Stratégie de segmentation

- **Optimisation de k Par Elbow Method**







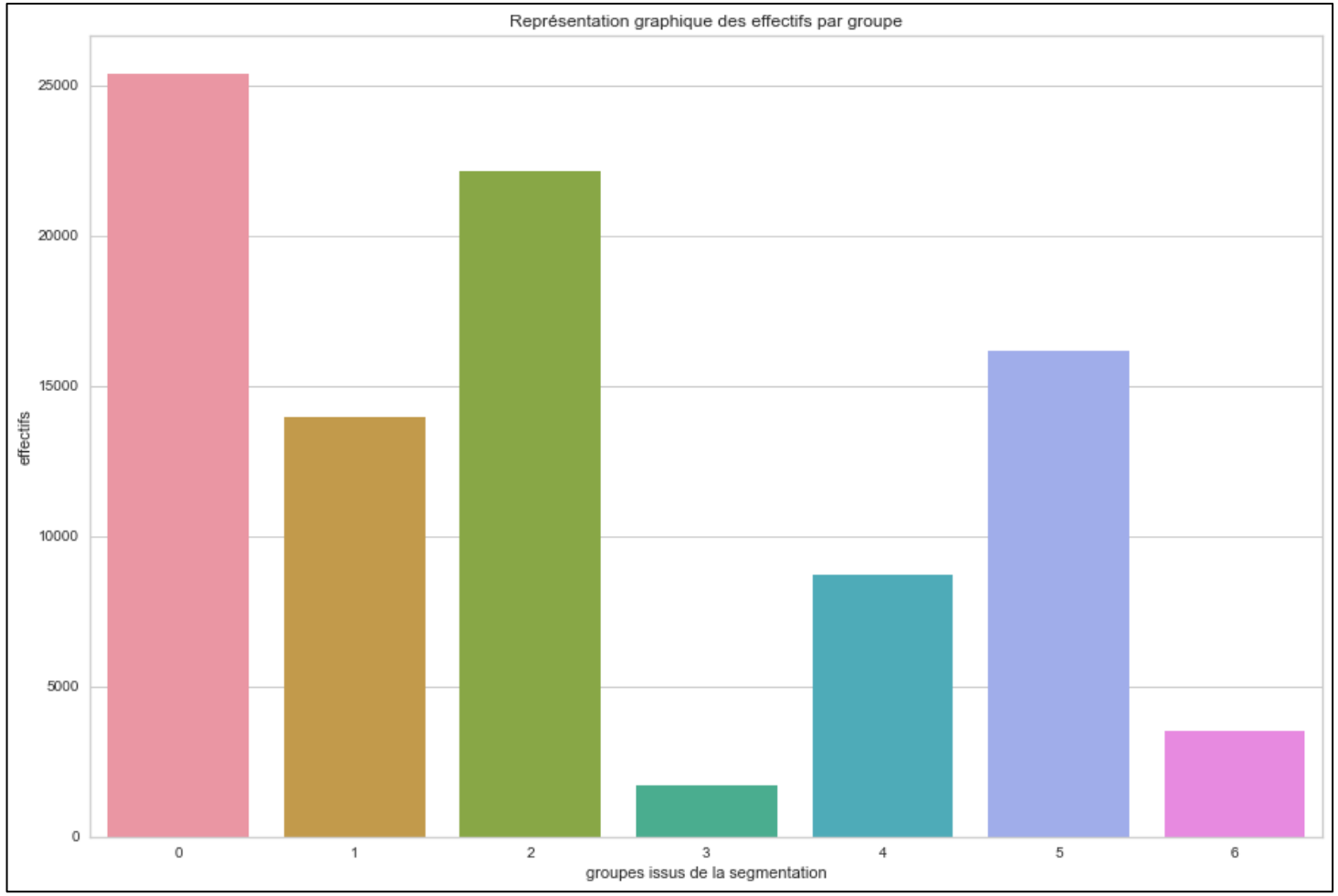
K optimisé = moyenne des 3 k obtenus

K_opt = 7

Analyse descriptive des segments

Analyse descriptive des segments

- **Effectifs**




```
count(data["group"])
```

```
Counter({4: 8727, 2: 22125, 0: 25370, 1: 13953, 5: 16142, 6: 3493, 3: 1682})
```

Analyse descriptive des 7 segments

- **Persona du 1er segment**

  - Les plus nombreux (28% des effectifs)
  - Comportement de paiement instables
  - Peu de commandes par client mais fort potentiel de dépense
  - Exigeants sur les délais de livraison mais satisfaits
  - Ne prennent pas le temps de s'impliquer dans la rédaction d'avis détaillés

- **Persona du 2ème segment**

  - 4ème groupe en terme d'effectifs (15%)
  - Comportement de paiement instable
  - Clients qui dépensent le plus par commande et 2.7% d'entre elles concernent plusieurs articles
  - Exigeants sur les délais de livraison et capables d'y mettre le prix
  - Clients insatisfaits qui prennent le temps de laisser des avis souvent avant réception

Analyse descriptive des 7 segments

- **Persona du 3ème segment**

  - 2ème groupe en terme d'effectifs
  - Achètent exclusivement en local
  - Ne sont pas influencés par le niveau de détails des annonces
  - Dépensent peu mais 4.3% d'entre elles concernent plusieurs articles
  - Exigeants sur les délais de livraison avec des frais peu élevés
  - Clients satisfaits, réactifs dans la rédaction des avis lorsqu'ils en laissent.

- **Persona du 4ème segment**

  - Représente seulement 2% des effectifs (le plus petit segment)
  - 6% des clients ont plusieurs comptes client
  - Donnent de l'importance aux détails des annonces
  - Comportement de paiement stable
  - Exigeant sur les délais de livraison
  - Clients satisfaits qui laissent des avis courts mais qui sont réactifs dans leur rédaction

Analyse descriptive des 7 segments

- **Persona du 5ème segment**

  - 10% des effectifs
  - Nombre de comptes client par client le plus faible et apprécient les annonces détaillées.
  - Capacité de dépense élevée
  - Nombre de commande par client inférieur à la moyenne mais 2.9% d'entre elles concernent plusieurs articles
  - Clients insatisfaits qui rédigent des avis complets rapidement après réception

- **Persona du 6ème segment**

  - 3ème groupe en terme d'effectifs (18%)
  - Dépensent peu mais acceptent de forts taux de frais de livraison
  - Paient exclusivement par boleto et ont le comportement de paiement le plus stable
  - Acceptent les délais de livraison les plus longs du panel
  - Clients satisfaits et réactifs dans la rédaction d'avis

Analyse descriptive des 7 segments

- **Persona du 7ème segment**

  - Seulement 4% des effectifs
  - Nombre de comptes client par client le plus élevé (r=1.06)
  - 7.3% des clients ont des comptes multiples (jusqu'à 17 comptes)
  - Dépensent peu mais le nombre de commandes par client et le nombre d'articles par commande sont les plus élevés
  - Paient exclusivement en voucher et acceptent les proportions de frais de livraison les plus élevés
  - Clients plutôt satisfaits mais mettent plus de temps que les autres à laisser leurs notes et avis.

Stabilité de la segmentation

Stabilité de la segmentation

- **Stratégie**

  - Par calcul et suivi du ARI score (Adjusted Rand Index score)
    - Calcul une mesure de similitude entre 2 segmentation par comparaison des clusters pour chaque paire d'échantillons
    - Comparer la segmentation prédite et la segmentation réelle pour une même population
    - Entre 0 (segmentation aléatoire) et 1 (segmentations identiques)
    - Suivi de l'évolution du ARI par mois sur toute la période disponible

  - Modèle kmeans pour les prédictions
    - Nombre de composantes principales fixe = 15 premières
    - Paramètre k fixe = 7
    - Entraîné sur les 6 premiers mois de données

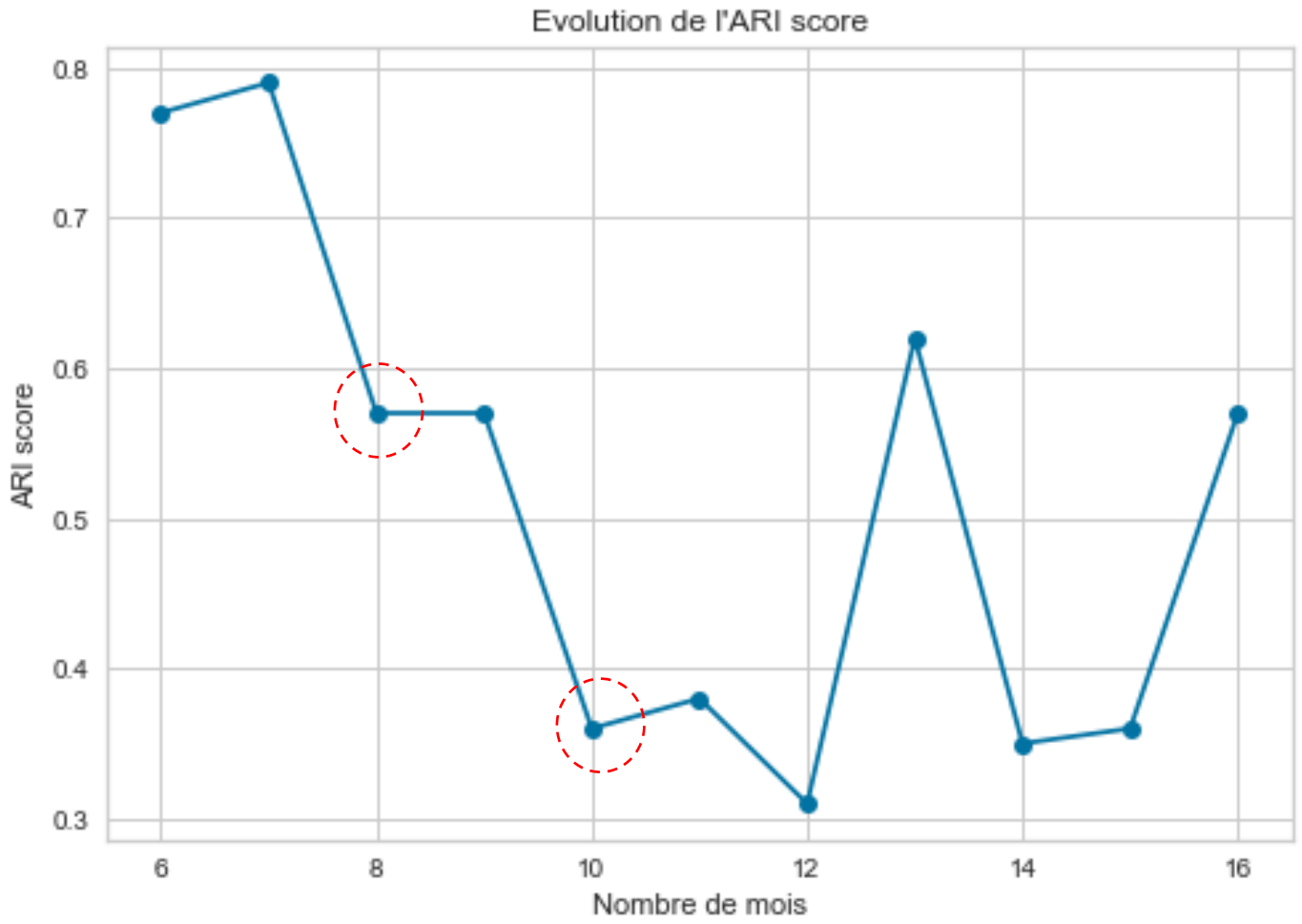
  - Les populations
    - Tous les mois entre le 7ème mois et la fin de la période
    - 11 périodes
    - Pour les segmentations en temps réel, le modèle kmeans sera réentraîné sur les nouvelles données

Stabilité de la segmentation

- **Résultats**

```
for labels in labels_all:
    print(len(labels))
```

```
5086
7323
10668
13767
17415
21344
25508
29765
36105
42209
48405
```



- Instabilité du modèle au bout de 2mois
- Au vu des contraintes métier, je propose une mise à jour du modèle tous les 4 mois

Conclusion

Conclusions

- **Feature engineering:**

  - Large exploitation des données fournies
  - Création de features originales qui traduisent un comportement client :
    - Score de comportement de paiement
    - Degrés de détails des annonces
    - Le type de morphologie des avis laissés
    - Impact des frais de livraison dans l'acte d'achat
  - Ces features ont permit de segmenter distinctement les clients

- **La segmentation:**

  - Les 7 clusters proposés ont assez de différences pour justifier ce clustering
  - Le modèle serait à actualiser tous les 2 ou 4 mois selon les contraintes client.

- **Voies d'amélioration:**

  - Faire varier le k et voir si les features créées restent pertinentes
  - Essayer d'améliorer la stabilité du modèle en jouant sur le jeu de données d'apprentissage

Merci pour votre attention