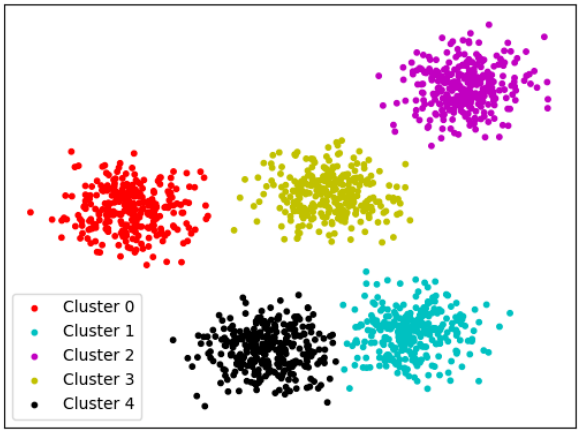
Université Abdelmalek Essaadi

Faculté des Sciences et Techniques de Tanger

Département Génie Informatique

Atelier 3 : Clustering



Encadré par :
Prof. Lotfi ELAACHAK

Elaboré par :
ELHANSALI Mouaad

Cycle d'ingénieur Logiciels et Systèmes Intelligents
S3 - 2024/2025

Table des matières

1 Exploration des Données

Dans cette étape, nous avons commencé par explorer les données du Data Set. Nous avons utilisé la bibliothèque `pandas` pour charger et afficher les premières lignes du jeu de données.

```
import pandas as pd

# Charger le DataFrame
df = pd.read_csv('chemin_vers_votre_fichier.csv')

# Afficher les premi res lignes du DataFrame
df.head()
```

```
Premieres lignes des donnees:
  CUST_ID      BALANCE  BALANCE_FREQUENCY  PURCHASES  ONEOFF_PURCHASES  \
0  C10001    40.900749           0.818182      95.40              0.00
1  C10002  3202.467416           0.909091       0.00              0.00
2  C10003  2495.148862           1.000000     773.17            773.17
3  C10004  1666.670542           0.636364    1499.00           1499.00
4  C10005   817.714335           1.000000      16.00             16.00

   INSTALLMENTS_PURCHASES  CASH_ADVANCE  PURCHASES_FREQUENCY  \
0                    95.4      0.000000             0.166667
1                     0.0   6442.945483             0.000000
2                     0.0      0.000000             1.000000
3                     0.0    205.788017             0.083333
4                     0.0      0.000000             0.083333

   ONEOFF_PURCHASES_FREQUENCY  PURCHASES_INSTALLMENTS_FREQUENCY  \
0                    0.000000                          0.083333
1                    0.000000                          0.000000
2                    1.000000                          0.000000
3                    0.083333                          0.000000
4                    0.083333                          0.000000

   CASH_ADVANCE_FREQUENCY  CASH_ADVANCE_TRX  PURCHASES_TRX  CREDIT_LIMIT  \
0                0.000000                 0              2        1000.0
1                0.250000                 4              0        7000.0
2                0.000000                 0             12        7500.0
3                0.083333                 1              1        7500.0
4                0.000000                 0              1        1200.0

      PAYMENTS  MINIMUM_PAYMENTS  PRC_FULL_PAYMENT  TENURE
0   201.802084        139.509787          0.000000      12
1  4103.032597       1072.340217          0.222222      12
2   622.066742        627.284787          0.000000      12
3     0.000000               NaN          0.000000      12
4   678.334763        244.791237          0.000000      12

Résumé statistique des donnees:
            BALANCE  BALANCE_FREQUENCY     PURCHASES  ONEOFF_PURCHASES  \
count   8950.000000        8950.000000   8950.000000       8950.000000
mean    1564.474828           0.877271   1003.204834        592.437371
std     2081.531879           0.236904   2136.634782       1659.887917
min        0.000000           0.000000      0.000000          0.000000
25%      128.281915           0.888889     39.635000          0.000000
50%      873.385231           1.000000    361.280000         38.000000
75%     2054.140036           1.000000   1110.130000        577.405000
max    19043.138560           1.000000  49039.570000      40761.250000

       INSTALLMENTS_PURCHASES  CASH_ADVANCE  PURCHASES_FREQUENCY  \
count             8950.000000   8950.000000          8950.000000
mean               411.067645    978.871112             0.490351
std                904.338115   2097.163877             0.401371
min                  0.000000      0.000000             0.000000
25%                  0.000000      0.000000             0.083333
50%                 89.000000      0.000000             0.500000
75%                468.637500   1113.821139             0.916667
max              22500.000000  47137.211760             1.000000

       ONEOFF_PURCHASES_FREQUENCY  PURCHASES_INSTALLMENTS_FREQUENCY  \
count                 8950.000000                       8950.000000
mean                     0.202458                          0.364437
std                      0.298336                          0.397448
min                      0.000000                          0.000000
25%                      0.000000                          0.000000
50%                      0.083333                          0.166667
75%                      0.300000                          0.750000
max                      1.000000                          1.000000

       CASH_ADVANCE_FREQUENCY  CASH_ADVANCE_TRX  PURCHASES_TRX  CREDIT_LIMIT  \
count             8950.000000       8950.000000    8950.000000   8949.000000
```

FIGURE 1 – Affichage des premières lignes du Data Set.

2 Visualisation des Données

Dans cette étape, nous avons exploré les données en utilisant des visualisations afin de mieux comprendre la distribution des caractéristiques et les relations entre elles. Nous avons utilisé un `scatter matrix` pour afficher les relations entre les différentes variables du jeu de données.

```
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt

# Visualisation du scatter matrix
scatter_matrix(df_numeric_filled, alpha=0.5, figsize=(10, 10), diagonal=
    'hist')
plt.show()
```

Le scatter matrix permet d'observer les corrélations et les distributions des différentes caractéristiques du dataset. Les graphiques sur la diagonale montrent l'histogramme de chaque caractéristique, tandis que les autres graphiques illustrent les relations entre les paires de variables.

3 Réduction de Dimension avec PCA et t-SNE

Une fois les données nettoyées, nous avons appliqué deux techniques de réduction de dimension : la PCA et le t-SNE. La PCA a réduit les données à deux composantes principales, tandis que t-SNE a permis une réduction de dimensionnalité non linéaire.

3.1 PCA

Le code pour appliquer PCA est le suivant :

```
from sklearn.decomposition import PCA

# PCA
pca = PCA(n_components=2)
pca_result = pca.fit_transform(df_numeric_filled)
```

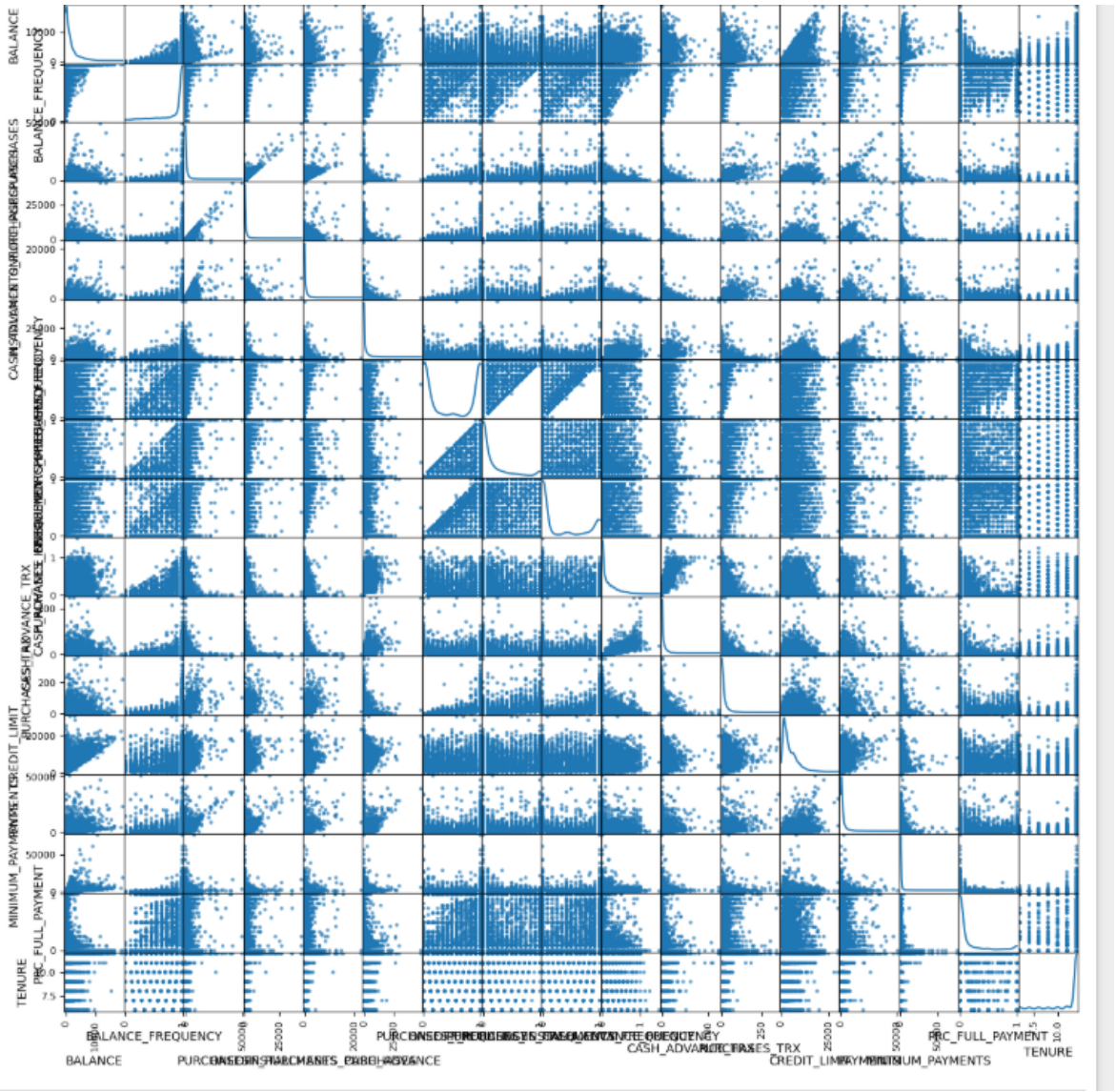


FIGURE 2 – Scatter Matrix représentant les relations entre les différentes variables.

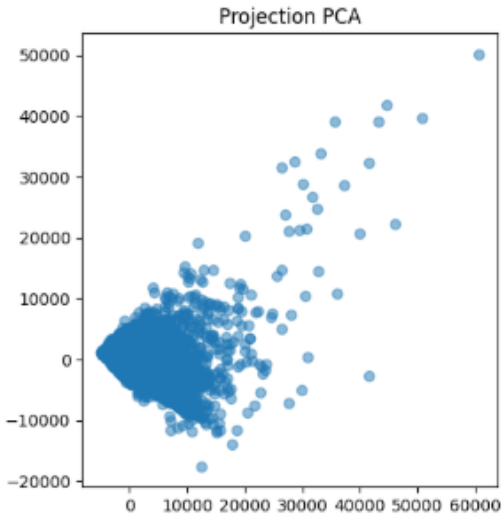


FIGURE 3 – Projection des données avec PCA.

3.2 t-SNE

Ensuite, nous avons appliqué t-SNE pour observer une réduction de dimensionnalité non linéaire.

```
from sklearn.manifold import TSNE

# t-SNE
tsne = TSNE(n_components=2, random_state=42)
tsne_result = tsne.fit_transform(df_numeric_filled)
```

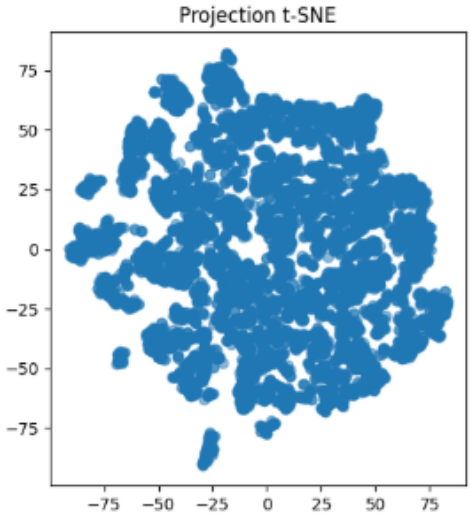


FIGURE 4 – Projection des données avec t-SNE.

4 Clustering avec K-Means

Dans cette section, nous appliquons l'algorithme de clustering K-Means sur les données réduites à deux dimensions, obtenues à partir des techniques de réduction de dimensionnalité PCA et t-SNE. Nous déterminerons également le nombre optimal de clusters à l'aide de la méthode du Elbow et visualiserons les résultats des clusters.

4.1 Application de K-Means sur les résultats de PCA et t-SNE

Nous appliquons l'algorithme K-Means sur les résultats obtenus après la réduction de dimensionnalité par PCA et t-SNE. Nous avons choisi un nombre initial de 3 clusters, mais ce nombre sera ajusté après l'analyse de la méthode du Elbow.

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# K-Means sur PCA
```

```
kmeans_pca = KMeans(n_clusters=3, random_state=42)  # Exemple avec 3 clusters
kmeans_pca.fit(pca_result)
pca_clusters = kmeans_pca.labels_

# K-Means sur t-SNE
kmeans_tsne = KMeans(n_clusters=3, random_state=42)  # Exemple avec 3 clusters
kmeans_tsne.fit(tsne_result)
tsne_clusters = kmeans_tsne.labels_

# Visualisation des clusters PCA
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.scatter(pca_result[:, 0], pca_result[:, 1], c=pca_clusters, cmap='viridis')
plt.title('Clusters PCA')

# Visualisation des clusters t-SNE
plt.subplot(1, 2, 2)
plt.scatter(tsne_result[:, 0], tsne_result[:, 1], c=tsne_clusters, cmap='viridis')
plt.title('Clusters t-SNE')

plt.show()
```

Listing 1 – Application de K-Means sur les résultats PCA et t-SNE

4.2 Méthode du Elbow pour déterminer le nombre de clusters

Pour déterminer le nombre optimal de clusters, nous utilisons la méthode du Elbow. Cette méthode consiste à calculer l'inertie pour différents nombres de clusters et à observer le point où l'inertie cesse de diminuer rapidement. Nous appliquons cette méthode à la fois sur les résultats PCA et t-SNE.

```
# M thode du Elbow pour PCA
inertia_pca = []
for k in range(1, 11):
    kmeans_pca = KMeans(n_clusters=k, random_state=42)
    kmeans_pca.fit(pca_result)
    inertia_pca.append(kmeans_pca.inertia_)

# M thode du Elbow pour t-SNE
inertia_tsne = []
for k in range(1, 11):
```

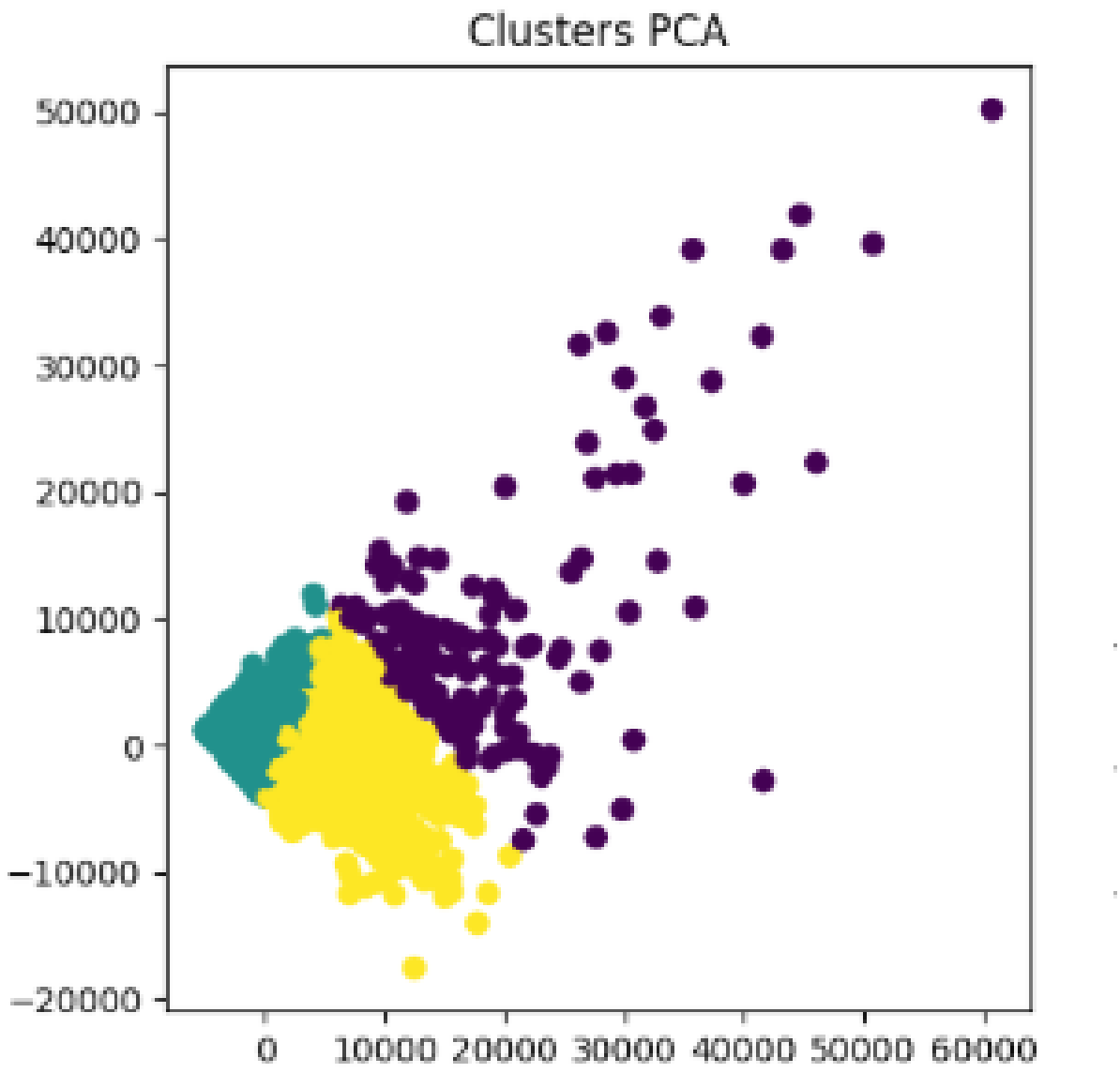


FIGURE 5 – Clusters obtenus avec PCA

```
    kmeans_tsne = KMeans(n_clusters=k, random_state=42)
    kmeans_tsne.fit(tsne_result)
    inertia_tsne.append(kmeans_tsne.inertia_)

# Tracer la courbe Elbow pour PCA
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(range(1, 11), inertia_pca, marker='o')
plt.title('M thode du Elbow pour PCA')
plt.xlabel('Nombre de clusters')
plt.ylabel('Inertie')

# Tracer la courbe Elbow pour t-SNE
```

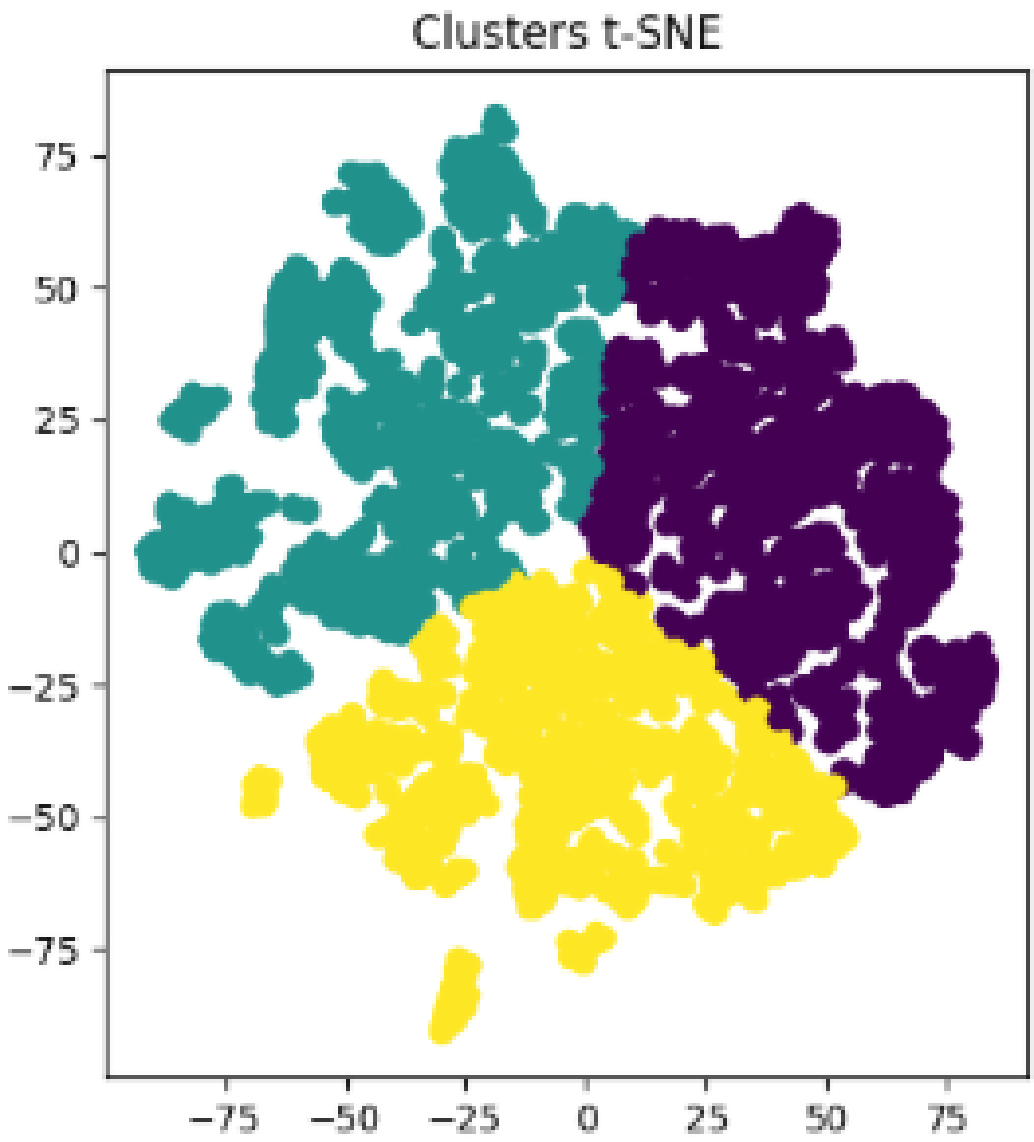


FIGURE 6 – Clusters obtenus avec t-SNE

```
plt.subplot(1, 2, 2)
plt.plot(range(1, 11), inertia_tsne, marker='o')
plt.title('M thode du Elbow pour t-SNE')
plt.xlabel('Nombre de clusters')
plt.ylabel('Inertie')

plt.show()
```

Listing 2 – Méthode du Elbow pour déterminer le nombre de clusters

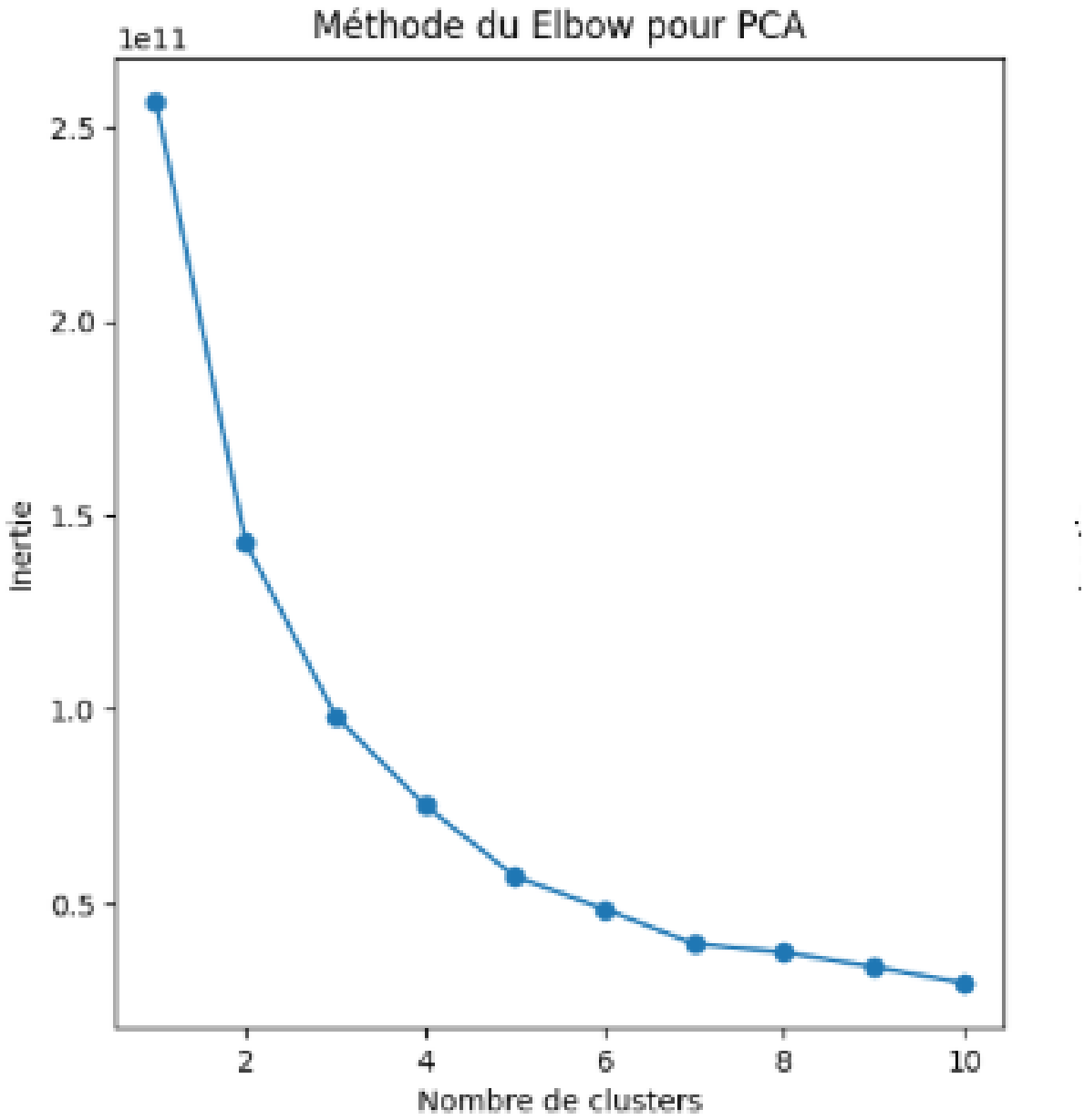


FIGURE 7 – Méthode du Elbow pour PCA

4.3 Interprétation des résultats

Les résultats des clusters formés par K-Means sur les données réduites par PCA et t-SNE sont visibles dans les graphiques ci-dessus. Les points sont colorés en fonction des clusters auxquels ils appartiennent. En observant la courbe du Elbow, nous pouvons déterminer le nombre optimal de clusters. Cela nous aidera à comprendre la structure sous-jacente des données et à choisir la configuration la plus appropriée pour le clustering.

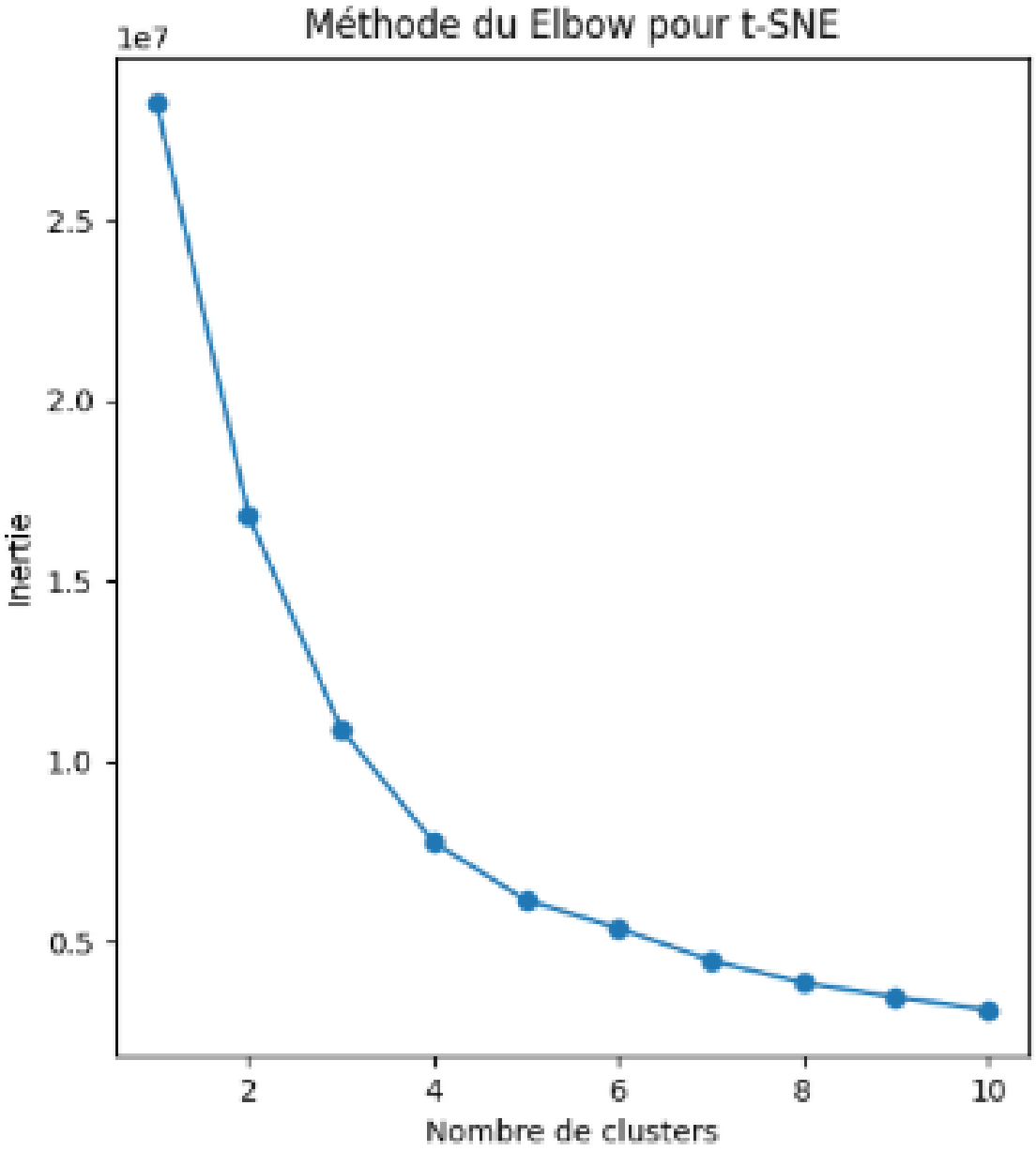


FIGURE 8 – Méthode du Elbow pour t-SNE

5 Interprétation des Résultats des Modèles PCA et t-SNE

Dans cette section, nous interprétons les résultats obtenus des deux modèles de réduction de dimensionnalité : l'Analyse en Composantes Principales (PCA) et l'Embedding Stochastique de Voisins (t-SNE).

5.1 PCA (Analyse en Composantes Principales)

L'Analyse en Composantes Principales (PCA) est une méthode linéaire qui vise à réduire la dimensionnalité des données tout en préservant autant que possible la variance. Dans notre cas, nous avons réduit les données à deux dimensions. Voici les points importants à retenir lors de l'interprétation des résultats de PCA :

— **Réduction de la dimensionnalité :** La réduction des données à deux dimensions permet de visualiser la structure principale des données tout en conservant une grande partie de la variance.

— **Variance expliquée :** Si la variance expliquée par les deux premières composantes principales est élevée (près de 100%), cela indique que PCA a bien capté la structure des données.

— **Interprétation du graphique :** Si les points sont bien séparés, cela peut indiquer une structure sous-jacente dans les données. Si les points sont très proches, cela peut suggérer une absence de structure claire.

5.2 t-SNE (t-Distributed Stochastic Neighbor Embedding)

Le t-SNE est une méthode non linéaire qui cherche à préserver les relations locales entre les points dans un espace de faible dimension. Voici les points clés de l'interprétation des résultats t-SNE :

— **Structure locale :** t-SNE maintient les relations de proximité locales entre les points, ce qui permet de visualiser les groupes ou clusters de données.

— **Séparation des clusters :** Si t-SNE montre une séparation nette entre les groupes de points, cela peut suggérer des clusters ou structures distinctes dans les données.

— **Non-linéarité :** Contrairement à PCA, qui est linéaire, t-SNE est mieux adapté pour capturer des relations non linéaires et des structures complexes dans les données.

5.3 Comparaison des deux modèles

- **PCA** aide à comprendre la variance globale des données, mais peut ne pas capturer des structures complexes non linéaires. - **t-SNE** est plus efficace pour visualiser des structures locales et des clusters dans des données non linéaires.

Ainsi, PCA permet d'obtenir une vue d'ensemble de la structure des données, tandis que t-SNE aide à explorer les relations locales et à identifier des groupes ou clusters potentiels.