Exercice 2 – Techniques de Fouille de Données

Mohamed BARBYCH

Code source : https://github.com/MohamedBarbych/TD-Kmeans-cc

1. Clustering avec K-Means

a) Explication de l'algorithme K-Means

L'algorithme K-Means est une méthode de regroupement (clustering) non supervisée qui permet de diviser un ensemble de données en K groupes homogènes appelés clusters.
Il vise à minimiser la distance (inertie) entre les points et le centre de leur cluster.

Étapes de l'algorithme :
1. Initialiser K centres de clusters (centroïdes) aléatoirement.
2. Assigner chaque point de données au centre le plus proche (souvent via la distance euclidienne).
3. Recalculer les nouveaux centroïdes en prenant la moyenne des points dans chaque cluster.
4. Répéter les étapes 2 et 3 jusqu'à convergence (plus aucun changement).

Ce processus est utile pour segmenter les clients selon leurs comportements d'achat.

b) Choix du nombre K optimal

Le choix de K est crucial. Deux méthodes courantes :
- La méthode du coude (Elbow Method) : observer la diminution de l'inertie intra-cluster selon K.
- Le score de silhouette : mesure la qualité de séparation entre clusters.

Un **K** mal choisi peut entraîner un sur- ou sous-clustering des données.

c) Application de K-Means (K=3)

Nous considérons un encodage One-Hot des produits financiers pour obtenir une matrice binaire de dimension (clients x produits). Chaque client est représenté par un vecteur de présence (1) ou absence (0) pour chaque produit :

Client	Prêt Pers.	Épargne	Crédit	Hypothèque	Assurance	Auto	Courant
Client 1	1	1	1	0	0	0	0
Client 2	0	0	0	1	1	0	0
Client 3	0	0	0	0	1	1	1
Client 4	0	1	1	0	0	0	0
Client 5	1	1	0	0	1	0	0
Client 6	0	0	0	1	0	0	1

Étape 1 : Initialisation

Nous choisissons aléatoirement 3 clients comme centroïdes initiaux :

- C1 = Client 1 → (1,1,1,0,0,0,0)
- C2 = Client 2 → (0,0,0,1,1,0,0)
- C3 = Client 3 → (0,0,0,0,1,1,1)

Étape 2 : Calcul des distances euclidiennes

On calcule pour chaque client sa distance aux 3 centroïdes initiaux. Exemple de

formule pour Client i et centroïde j :
$d(i,j) = \sqrt{((xi1 - cj1)^2 + (xi2 - cj2)^2 + ... + (xi7 - cj7)^2)}$

Voici un extrait des calculs (valeurs arrondies) :

Client	d(C1)	d(C2)	d(C3)	Cluster assigné
Client 1	0.0	2.45	2.45	C1
Client 2	2.45	0.0	2.0	C2
Client 3	2.45	2.0	0.0	C3
Client 4	1.0	2.45	2.45	C1
Client 5	1.73	2.0	2.45	C1
Client 6	2.0	1.0	1.73	C2

Étape 3 : Mise à jour des centroïdes

- Nouveau C1 = moyenne de Clients 1, 4, 5 :
  → Moyenne : (0.67, 1.0, 0.67, 0.0, 0.33, 0.0, 0.0)

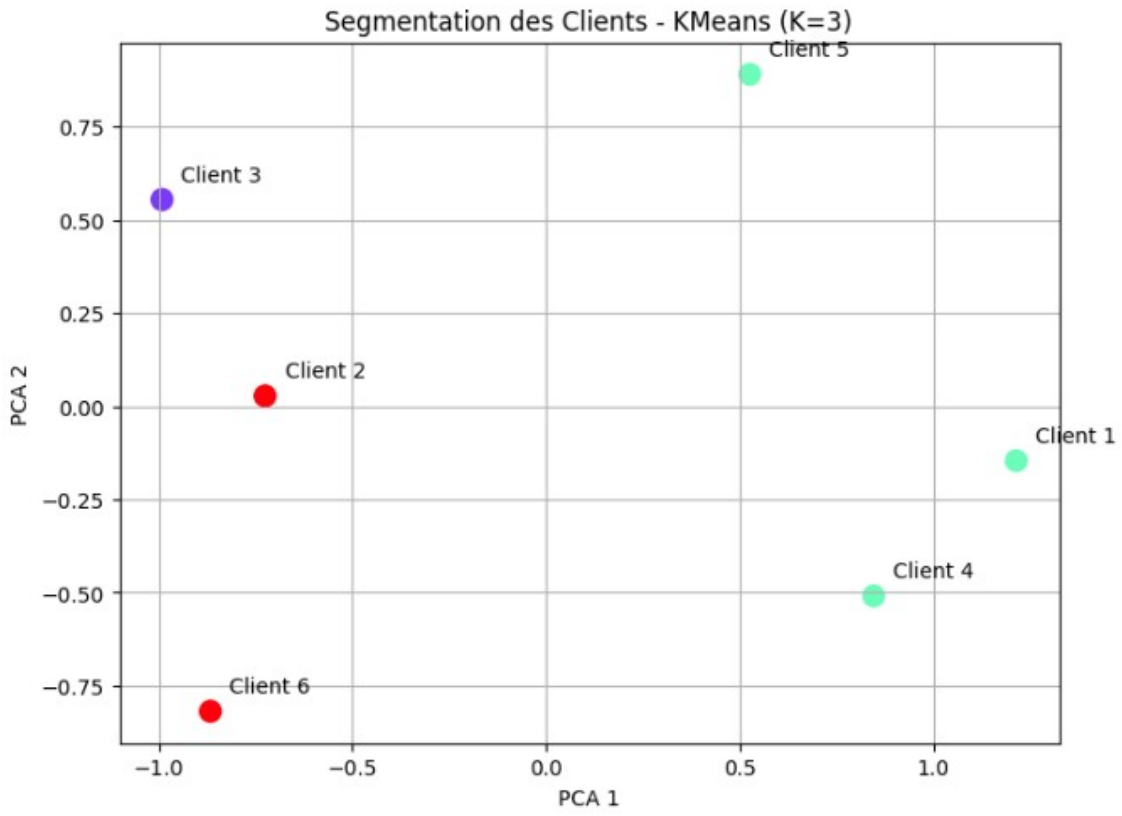
- Nouveau C2 = moyenne de Clients 2, 6 :
  → Moyenne : (0.0, 0.0, 0.0, 1.0, 0.5, 0.0, 0.5)

- Nouveau C3 = Client 3 seul :
  → Inchangeable : (0,0,0,0,1,1,1)

Étape 4 : Résultat final

Affectation définitive des clients dans les clusters :

- Cluster 1 : Clients 1, 4, 5 → Produits : Prêt personnel, épargne, carte de crédit
- Cluster 2 : Clients 2, 6 → Produits : Hypothèque, Assurance, Compte courant
- Cluster 3 : Client 3 → Produits : Crédit automobile, Assurance vie, Compte courant



Cela permet à l'entreprise de personnaliser les campagnes marketing et les recommandations selon des profils bien différenciés.

2. Règles d'Association avec l'algorithme Apriori

L'algorithme Apriori permet d'extraire les combinaisons fréquentes de produits achetés ensemble dans des transactions.

On applique :
- un support minimal de 40%
- une confiance minimale de 60%

Résultats :
- Règle 1 : {Compte Épargne, Carte de Crédit} → {Prêt Personnel}, confiance = 0.75
- Règle 2 : {Carte de Crédit} → {Compte Épargne}, confiance = 1.00

- Règle 3 : {Prêt Personnel} → {Compte Épargne}, confiance = 1.00

Interprétation : ces règles permettent d'identifier des opportunités de vente croisée.

Applications pour l'entreprise

- Ciblage marketing plus intelligent selon les segments
- Recommandations personnalisées de produits
- Optimisation des campagnes promotionnelles
- Fidélisation via des offres adaptées aux profils clients