Clustering de Données : Ensemble de Données Produit

1. Ensemble de données Produit

Nous allons durant cette séance utiliser l'ensemble de données *Data_Produit_QF.csv* qui contient des données sur l'achat d'un produit qui a été proposé à des clients. La variable *Produit* indique pour chaque client s'il a ou non acheté le produit.

Caractéristiques de l'ensemble de données :

- Instances : 600, chacune décrivant un client
- Nombre de variables : 13
- Variable de classe : *Produit*
- Valeurs manquantes : aucune

Le dictionnaire des données ci-dessous indique pour chacune des 13 variables :

- son libellé (nom),
- son type (entier, décimal, booléen, catégoriel, ordinal, date, etc.),
- sa description (sémantique),
- son domaine de valeurs (intervalle [minimum, maximum] pour les variables numériques ou liste de valeurs pour les variables modales/catégorielles).

Dictionnaire des données

Variable	Type	Description	Domaine de valeurs
ID	Entier	Numéro identifiant du client	[12101, 12400]
Age	Entier	Age en années	[18, 67]
Sexe	Catégoriel	Sexe	Homme, Femme
Habitat	Catégoriel	Type d'habitat	Centre_Ville, Petite_Ville, Rural, Banlieue
Revenus	Entier	Revenus annuels en dollars US	[60392, 505040]
Marie	Booléen	Statut marital	Oui, Non
Enfants	Entier	Nombres d'enfants	[0, 3]
Voiture	Booléen	Possède une voiture	Oui, Non
Compte_Epargne	Booléen	Possède un compte épargne	Oui, Non
Compte_Courant	Booléen	Possède un compte courant	Oui, Non
Emprunt	Booléen	Emprunt en cours	Oui, Non
Quotient_Familial	Entier	Rapport entre revenus et nombre d'enfants	[20280, 492400]
Produit	Booléen	Client acquéreur du produit Variable de classe	Oui, Non

1.1. Objectif

L'objectif de cette application est d'essayer d'identifier parmi chacune des deux classes, Produit = Oui et Produit = Non, des sous-groupes de clients, c-à-d un cluster de clients, ayant des caractéristiques communes (âge moyen, nombre d'enfants, etc.) qui leur sont spécifiques, c-à-d qui sont différentes de celles autres sous-groupes de la même classe.

Pour cela, plusieurs algorithmes de clustering (*K-means*, *Agnes*, etc.) seront appliqués, avec pour objectif de comparer la création de différents clusterings, chacun correspondant à un nombre spécifique de clusters (paramètre *K*) pour un algorithme particulier (ex : *K-means* pour *K* = 4).

1.2. Chargement des données

☛ Chargez les données du fichier *Data_Produit_QF.csv* dans un data frame `produit`.

☛ Supprimez la variable `ID` du data frame `produit` avec l'opérateur de sélection `[,]`.

- Modifiez le type de la variable *Enfants* du data frame `produit` pour le type ordinal (valeurs catégorielles ordonnée) par la commande :

```
> produit$Enfants <- as.ordered(produit$Enfants)
```

Le type ordinal permettra lors des calculs de distance entre les instances de tenir compte de l'ordre dans les valeurs.

- Vérifiez le résultat de l'opération en affichant la classe et le mode de la variable `Enfants` à laide des fonction `class()` et `mode()`.
- Affichez les caractéristiques résumées du data frame `produit` par la fonction `str()`.
- Affichez les caractéristiques résumées des variables du data frame `produit` par la fonction `summary()`.

2. Matrice de distance

Nous allons utiliser la librairie *cluster* qui fournit un ensemble de méthodes pour le clustering.

- Chargez la librairie *cluster*.

Nous allons calculer une matrice de distance pour les instances du data frame `produit` en utilisant la fonction `daisy()` du package `cluster` qui permet de traiter les données hétérogènes (numériques, catégorielles, ordinales, etc.).

- Calculez la matrice de distance `dmatrix` à l'aide de la fonction `daisy()`.
- Affichez les informations résumées de la matrice de distance `dmatrix` à l'aide de la fonction `summary()`.

3. Clustering hiérarchique par agglomération

Nous allons utiliser la librairie *fpc* qui fournit un ensemble de méthodes pour le clustering hiérarchique.

- Installez et chargez la librairie *fpc*.

3.1. Construction du dendrogramme par l'algorithme Agnes

La fonction `agnes()` permet de réaliser un clustering hiérarchique par agglomération, dont le résultat est un dendrogramme, à partir d'une matrice de distance.

- Exécutez le clustering hiérarchique par `agnes()` en stockant le résultat dans un objet `agn` par la commande :

```
> agn <- agnes(dmatrix)
```

- Affichez le dendrogramme résultant `agn` à l'aide le la fonction `plot()` par la commande :

```
> plot(agn)
```

- Afin de délimiter par une bordure rouge les regroupements correspondant à 4 clusters dans le dendrogramme affiché, exécutez la commande :

```
> rect.hclust(agn, k=4, border="red")
```

3.2. Résultat pour un nombre de clusters donné

Le clustering obtenu, c-à-d le cluster d'affectation pour chaque instance, pour un nombre de clusters donné peut être obtenu par la fonction `cutree()`.

- Calculez le résultat pour 4 clusters, en le stockant dans un objet `agn4`, à partir de l'objet `agn` par la commande :

```
> agn4 <- cutree(agn, k=4)
```

L'objet `agn4` ainsi généré permet d'associer à chaque instance le numéro du cluster auquel cette instance est affectée.

- Affichez le numéro du cluster d'affectation (1, 2, 3 ou 4) de chaque instance du data frame `produit` par la commande :

```
> View(agn4)
```

- Affichez une table de contingence affichant pour chaque cluster le nombre d'instances de chaque classe par la commande :

```
> table(agn4, Produit)
```

- Affichez un histogramme d'effectifs pour chaque cluster (variable `agn4`) avec la proportion de

chaque classe en couleur (variable `produit$Produit`) :

```
> qplot(agn4, data=produit, fill=Produit)
```

3.3. Variation du nombre de clusters

Nous allons comparer les résultats obtenus pour différents nombres de clusters à partir du dendrogramme.

- Créez une boucle `for()` qui permet pour un nombre *K* de clusters variant de 3 à 8 de :
  - Générer un nouveau affichage du dendrogramme `agn` à l'aide le la fonction `plot()`.
  - Délimiter par une bordure rouge les regroupements correspondant à *K* clusters dans le dendrogramme affiché par la fonction `rect.hclust()`.
  - Calculer le résultat pour *K* clusters en le stockant dans un objet `agnK` par la fonction `cutree()`.
  - Créer la table de contingence affichant pour chaque cluster le nombre d'instances dans chaque classe.
  - Afficher l'histogramme d'effectifs pour chaque cluster avec la proportion de chaque classe en couleur.
- Quel est le nombre de clusters pour lequel la proportion d'instances de même classe dans chaque cluster est maximale ?

4. Clustering hiérarchique par division

4.1. Construction du dendrogramme par l'algorithme Diana

La fonction `diana()` permet de réaliser un clustering hiérarchique par division, dont le résultat est un dendrogramme, à partir d'une matrice de distance.

- Exécutez le clustering hiérarchique par `diana()` en stockant le résultat dans un objet `dia` par la commande :

```
> dia <- diana(dmatrix)
```

- Affichez le dendrogramme résultant `dia` à l'aide le la fonction `plot()` par la commande :

```
> plot(dia)
```

- Afin de délimiter par une bordure rouge les regroupements correspondant à 4 clusters dans le dendrogramme affiché, exécutez la commande :

```
> rect.hclust(dia, k=4, border="red")
```

4.2. Résultat pour un nombre de clusters donné

Le clustering obtenu, c-à-d le cluster d'affectation pour chaque instance, pour un nombre de clusters donné peut être obtenu par la fonction `cutree()`.

- Calculez le résultat pour 4 clusters, en le stockant dans un objet `dia4`, à partir de l'objet `dia` par la commande :

```
> dia4 <- cutree(dia, k=4)
```

L'objet `dia4` ainsi généré permet d'associer à chaque instance le numéro du cluster auquel cette instance est affectée.

- Affichez le numéro du cluster d'affectation (1, 2, 3 ou 4) de chaque instance du data frame `produit` par la commande :

```
> View(dia4)
```

- Affichez une table de contingence affichant pour chaque cluster le nombre d'instances de chaque classe par la commande :

```
> table(dia4, Produit)
```

- Affichez un histogramme d'effectifs pour chaque cluster (variable `dia4`) avec la proportion de chaque classe en couleur (variable `produit$Produit`) :

```
> qplot(dia4, data=produit, fill=Produit)
```

4.3. Variation du nombre de clusters

Nous allons comparer les résultats obtenus pour différents nombres de clusters à partir du dendrogramme.

- Créez une boucle `for()` qui permet pour un nombre *K* de clusters variant de 3 à 8 de :

- ☞ Générer un nouveau affichage du dendrogramme `dia` à l'aide le la fonction `plot()`.
- ☞ Délimiter par une bordure rouge les regroupements correspondant à *K* clusters dans le dendrogramme affiché par la fonction `rect.hclust()`.
- ☞ Calculer le résultat pour *K* clusters en le stockant dans un objet `diaK` par la fonction `cutree()`.
- ☞ Créer la table de contingence affichant pour chaque cluster le nombre d'instances dans chaque classe.
- ☞ Afficher l'histogramme d'effectifs pour chaque cluster avec la proportion de chaque classe en couleur.

☛ Quel est le nombre de clusters pour lequel la proportion d'instances de même classe dans chaque cluster est maximale ?