Chapitre 1 : Introduction à la fouille de données

1. Introduction

- Les données d'une entreprise doublent de volume chaque année. Elles peuvent être interrogées par des outils et langages qui ont prouvé leur efficacité.
- Par exemple, dans les SGBD relationnels, on utilise le langage SQL → cela suppose la connaissance des schémas de données, et du contenu général de la base.
- Cependant, le grand volume des données peut renfermer des connaissances que les outils classiques d'interrogation ne peuvent pas extraire
    - quel est le volume d'achat du client X, durant la période Y, quel est le meilleur client (max du volume d'achat, durée de fidélité...) → requête satisfaite par des outils classiques
    - quels sont les caractéristiques des clients qui rompent (change d'entreprise), comment savoir si un demandeur de prêt est solvable (peut rembourser le prêt?), … → requête non (ou difficilement) satisfaite par les outils classiques.

      → besoin d'outils et techniques spécifiques pour extraire des connaissances à partir de données
- On parle du KDD (knowledge discovery from data) ou d'ECD (extraction des connaissances à partir de données)
- La fouille de données (data mining - DM) est au cœur de l'ECD et en représente le moteur. C'est une ingénierie renfermant des outils, des techniques, des algorithmes, etc qu'elle puise dans
    - les statistiques et analyse de données
    - les bases de données
    - l'intelligence artificielle
- Le DM perment d'aboutir à des modèles (ex : fonction mathématique, règles logiques SI condition ALORS résultats) qui doivent être validés pour devenir des connaissances utilisables par l'être humain ou par les machines.

2. Définition de la fouille de données

Les définitions du data mining ne font pas parfois la différence entre le *data mining* qui est la fouille de données et le *KDD* ou *knowledge discovery from data* qu'on peut traduire par l'extraction des connaissances à partir des données. On cite les deux définitions suivantes

- Fayyad : « l'extraction de connaissances à partir des données est un processus non trivial d'identification de structures inconnues, valides et potentiellement exploitables dans les bases de données »
- Frawley : « extraction non triviale d'informations implicite, précédemment inconnue, et potentiellement utiles à partir des données ».

Les deux définitions précédentes laissent le champ ouvert aux techniques et applications du DM. On cite la classification, la régression, le clustering, les règles d'association, etc.

3. Historique de la fouille de données

La fouille de données est une évolution naturelle dans l'exploitation des données par les être humains en utilisant les ordinateurs. On peut résumer cette évolution dans les points suivants :

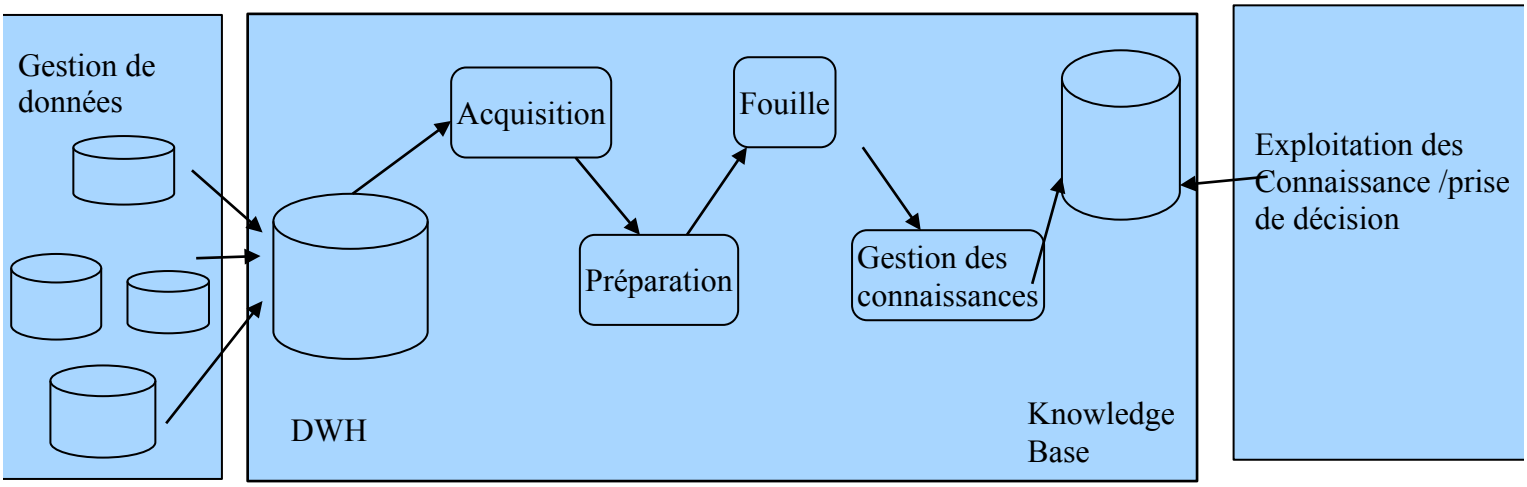
- Début de l'informatique : utilisation des ordinateurs pour les besoins de calcul
- traitement statistique des données et analyse de données : prémices du DM
- Fin des années 80 : exploitation du contenu des bases de données pour la recherche de règles d'association : utilisation du terme database mining
- 1989 : premier atelier sur la découverte de connaissances – proposition du terme Knowledge Discovery par Gregory Piatetsky-Shapiro
- 1995 : première conférence sur le data mining.

Par ailleurs, le DM a été influencé par

- l'explosion du volume de données produites et stockées
- la maturité des outils de reporting des données et l'évolution du besoin des utilisateurs (de la gestion de données vers la prise de décision)
- l'évolution de la relation avec les clients : vers le profiling des clients et la production orientée client.

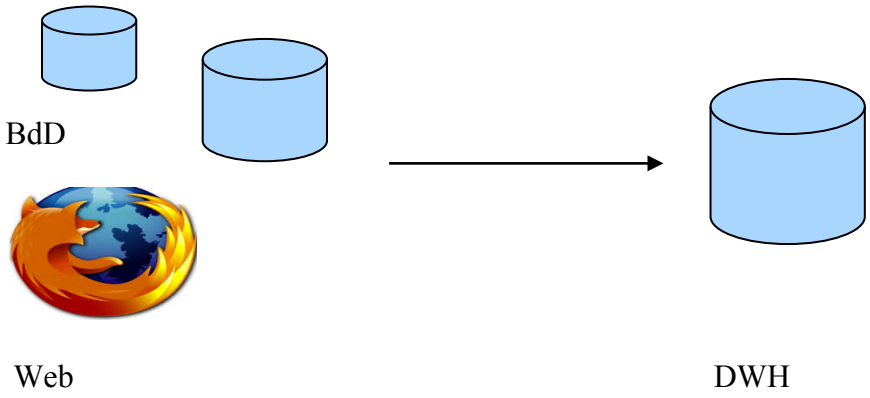
4. Processus de la fouille de données

- Le processus de fouille de données, ou généralement d'ECD se positionne en back-end au sein de l'entreprise ou des cabinets spécialisés.
- En fron-end, on trouve les activités de production des données en amont et de prise de décision en aval.
- Le schéma suivant (adapté de Zighed et Rakotomalala) résume les quatre étapes de l'ECD et le positionnement du DM au milieu comme maillon fort.



- Pré étape d'ECD : Alimentation de l'entrepôt de données / production de données

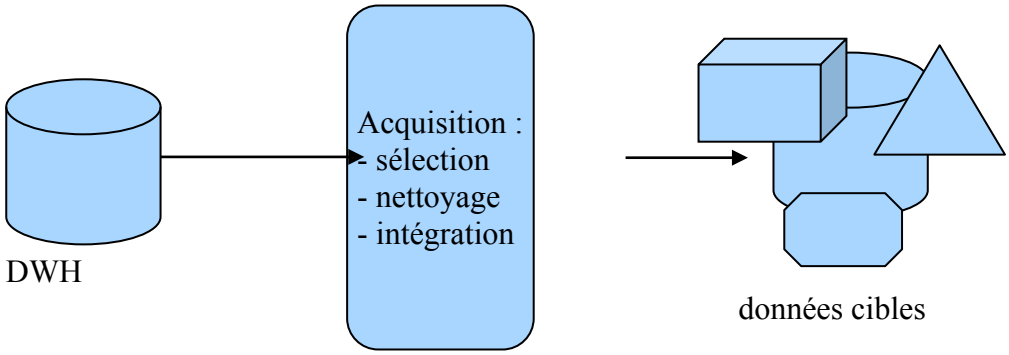
schéma :



Description : il s'agit de la partie en amont du front office qui consiste à alimenter l'entrepot de données (grande BdD) de l'entreprise à partir des bases de données de production, du Web ou d'autres sources (surveys, applications d'analyse, ...).

- Etape 1 : Acquisition de données

schéma :

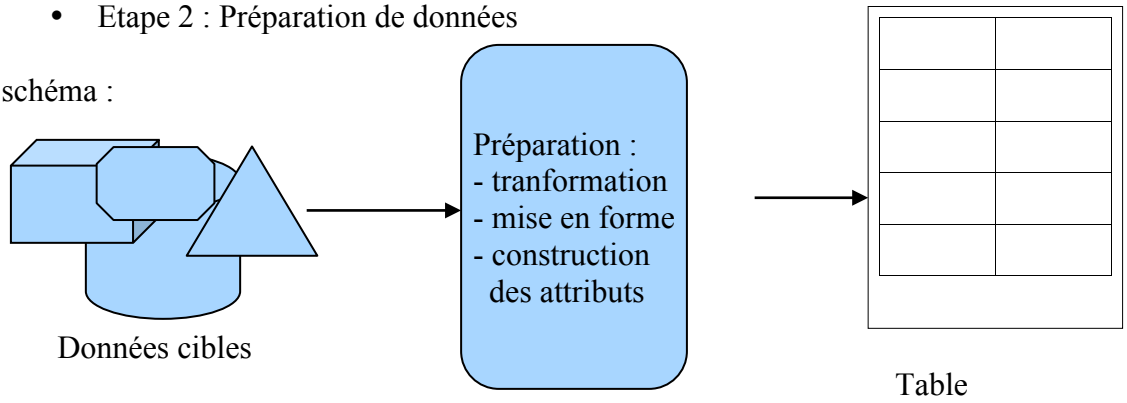


description :

- le processus d'ECD ne cible pas toutes les données de l'entreprise mais seulement celles qui serviront à résoudre le problème.
- L'acquisition permet de cibler les données utiles.
- On utilise des requêtes ad hoc (non pré définies), de l'échantionnage (sampling), etc...
- Il n'y a pas de limite de taille des données cibles
- Les données cibles peuvent nécessiter un nettoyage (élimination des attributs mal renseignés, erronés, etc).

- Etape 2 : Préparation de données

schéma :



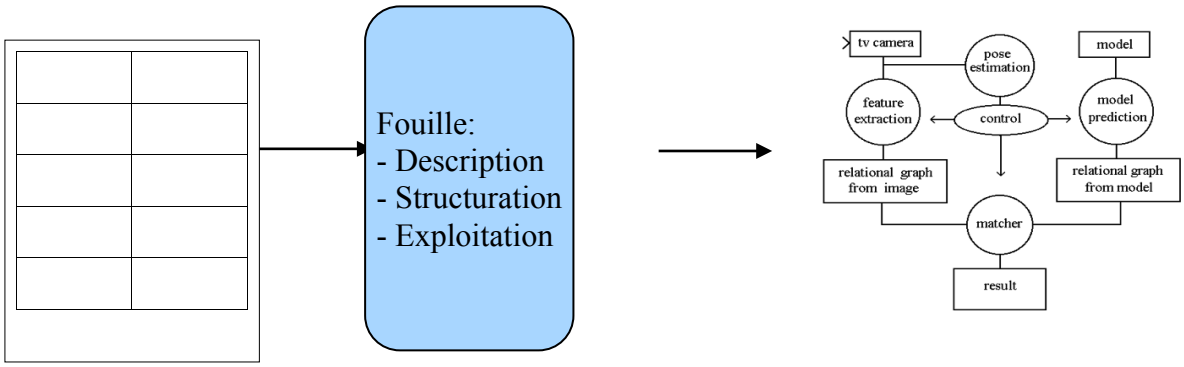
description :

- le plus souvent, les données exploitées par l'ECD doivent avoir une forme tabulaire (ligne/colonne).
- Si les données n'ont pas cette forme, elles sont transformées et adaptées.
- Parfois, même la forme tabulaire initiale nécessite une autre transformation (centrage, mise sous une forme binaire '0/1', etc.).
- La construction d'attributs inclut :
  - la réduction du volume de données par élimination des attributs inutiles
  - la transformation: par exemple passer d'un attribut continu (ex : température) à un attribut discret (intervalle).
  - la construction d'agrégats : il s'agit d'attributs obtenus à partir d'autres qui permettent d'effectuer des comparaisons (ex : remplacer le prix d'un appartement et la surface par le prix au mètre carré, remplacer la ville par la région, etc).

- A cette étape également, les données absentes ou erronées mais nécessaires sont traitées : on utilise différentes techniques comme
  - le remplacement de la valeur absente par la valeur la plus fréquente
  - l'estimation de la valeur absente à partir des valeurs existantes

- Etape 3 : Fouille de données

schéma



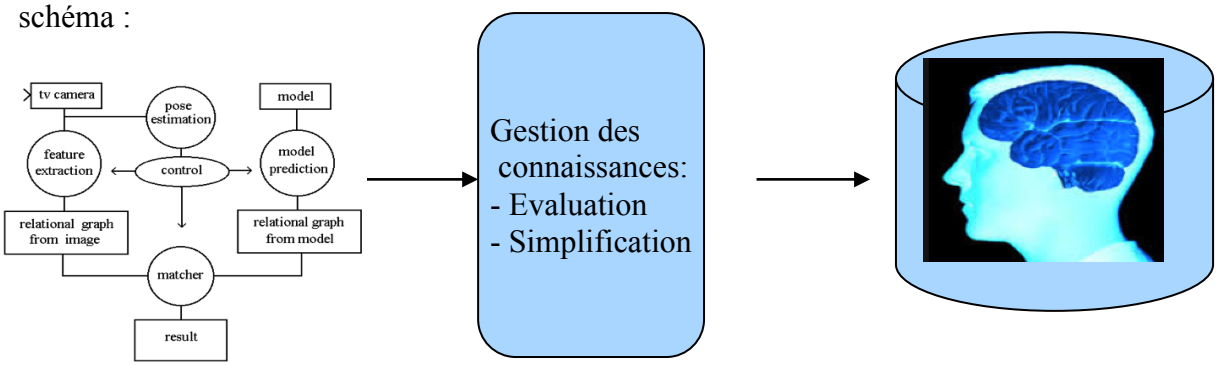
modèle/concept

description :

- phase de fouille à proprement parler
- met en œuvre différentes techniques, selon le problème à traiter (voir section suivante)
- permet d'obtenir des modèles et des concepts à valider.

- Etape 4 : Gestion des connaissances

schéma :



modèle / concept base de connaissances

description :

- Le modèle obtenu à partir de l'étape précédente doit être validé pour être utilisé dans des cas réel (ex : avant de pouvoir trancher sur un diagnotic d'un cas de maladie).
- On procède au calcul du taux d'erreur du modèle : si le taux d'erreur est accepté, on utilise le modèle. Sinon, le modèle n'est pas utilisé.

- Post étape d'ECD : exploitation des connaissances / prise de décision

schéma :

Décideur base de connaissances

description : une fois le modèle issu de la fouille de données validé ; le décideur l'utilise sur des données réelles pour prendre des décision selon le domaine. Ces décisions peuvent être par exemple

- d'accepter d'octroyer un prêt à un candidat,
- d'estimer le taux de vote et les résultats par candidat

- ...

5. Méthodes du data mining :

- selon le problème à traiter, il existe plusieurs méthodes de data mining.
- Ces méthodes peuvent être classés comme suit :

5.1. Visualisation et description :

- ces méthodes synthétisent et décrivent les données sous une forme visuelle qui permet une interprétation plus ou moins directe.
- elles se basent sur l'affichage d'indicateurs statistiques (moyennes, écart-type, médianes, modes, etc).
- Différentes solutions de visualisation peuvent être utilisées :

5. 1. 1. les courbes ou tableaux de statistiques
5. 1. 2. les histogrammes
5. 1. 3. les nuages de points
5. 1. 4. le graphe de contingence

5. 2. Classification et structuration :

- Dans ces méthodes, l'objectif est de classer un ensemble d'individus afin de mieux comprendre la réalité (simplification) ou pour d'autres fins (ex : identifier des groupes de clients ayant des profils similaires afin de les cibler par des messages communs).
- Ces méthodes relèvent de l'apprentissage non supervisé car l'utilisateur ne sait pas a priori quelles classes obtenir.
- Les méthodes utilisées sont les méthodes de classification automatique ou cluster analysis. (Voir chapitre 3).

5. 3. Explication et prédiction

- Dans ces méthodes, l'objectif est d'aboutir à un modèle d'explication d'un phénomène ou de prédiction. Des exemples sont (1) l'aide au diagnotic (si un patient est atteint ou pas d'une maladie), (2) la décision qu'un message est un spam, etc.
- parmi les méthodes, on cite
  - les arbres de décision
  - les réseaux de neurones
  - les réseaux bayésiens
  - les règles d'association
  - la régression
  - l'analyse discriminante

Note : les 4 premières méthodes seront traitées dans les chapitres de 4 à 7.

6. **Quelques domaines d'application**

- la fouille de données a rendu et rend encore des services dans différentes domaines d'applications, aucun domaine n'est a priori exclu
- Parmi ces domaines, on cite :
  - la gestion de la relation client : appelée aussi CRM. Parmi les applications :
    - profiling : connaître et regrouper les clients en profils pour mieux les cibler par des compagnes publicitaire → utilisation de la classification
    - marketing : regrouper les produits les plus fréquemment achetés ensembles dans les même stand → utilisation des règles d'association
  - la médecine : aide au diagnotic des malades → utilisation des arbres de décision, des réseaux bayésiens,...
  - les télécommunication
    - identfication des fraudes de cartes de crédit
  - la bureautique : identification des messages spam, aide contextuelle.
  - la politique : prédiction des résultats des élections

...

Support de cours en Fouilles de données
Chapitre 2 : Rappel des statistiques

1. Introduction

- Objet des statistiques :
  - Etude d'un ensemble d'individus sur lesquels on observe des caractéristiques appelées variables. Selon le nombre de variables, on distingue
    - Techniques simples résumant les caractéristiques d'une variable (moyenne, médiane, etc.), permettant de détecter les valeurs atypiques.
    - Techniques s'appliquant à deux variables ou plus (corrélation, nuage de points).

- Objectifs
  - Mieux connaître la population étudiée par l'explication des variables.
  - Prévoir le comportement des individus qui ne sont non encore observés.

2. Statistiques descriptives d'une variable

- Types de variables
  - *Variable quantitative :* les valeurs prises sont numériques. Ces valeurs peuvent être
    - discrètes : c'est à dire appartenant à une liste dénombrable. **Exemple** : le nombre de pannes d'une machine, le nombre des jours de travail.
    - continues : les valeurs prises ne peuvent pas être comptées et appartiennent à un intervalle. **Exemple** : la température, la moyenne annuelle d'un étudiant.
  - *Variable qualitative :* les valeurs prises sont des labels. Ces valeurs peuvent être
    - nominales : quand elles ne sont pas ordonnables. Exemple : la couleur.
    - ordinales : quand il est possible de les ordonner selon un sens : petit < moyen < grand ; faible < normal < puissant.

2.1. Cas d'une seule variable

- Notions :
  - **Moyenne** : la moyenne arithmétique est la somme des valeurs d'une variable quantitative, divisée par le nombre d'individus. Ex : la moyenne des âges 3, 12, 18 est 11.
  - **Médiane** : c'est la valeur qui sépare les valeurs d'une série statistique en deux. Ex : la médiane de la série 1 3 5 7 9 est 5.
  - **Mode** : il correspond à la valeur la plus fréquente. Ex : le mode de la série 2, 3, 3, 4, 7, 3, 2, 1, 3 est 3 avec l'effectif 4.
  - **Variance** : la variance sert à caractériser la dispersion des valeurs de la moyenne : variance de zéro → toutes les valeurs sont identiques, petite variance → les valeurs sont proches les unes des autres, variance élevée → celles-ci sont très écartées.

La formule de calcul de la variance (écart-type au carré) est la suivante.

$$\sigma^2 = \frac{\sum \left(X - \bar{X} \right)^2}{N}$$

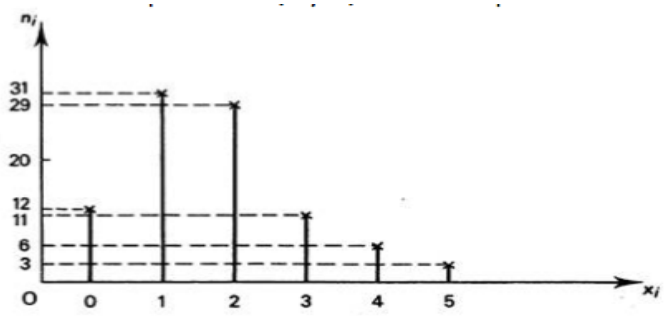
- **Ecart-type** : C'est la racine carrée de la variance. Il permet de mesurer l'écart entre les valeurs de la série avec la même grandeur que celle des valeurs.

- Représentations graphiques

Les données statistiques peuvent être représentées graphiquement pour une meilleure interprétation et mémorisation. Ces représentations sont parfois suffisantes à elles-mêmes pour visualiser une population. Il existe différentes représentations graphiques associées au cas mono-variable.

- *Diagramme en bâtons :* il permet de représenter des effectifs d'une variable discrète sur deux axes : en abscisses les individus observés et en ordonnées représente l'effectif de chaque individu. On parle aussi de nuage de points.

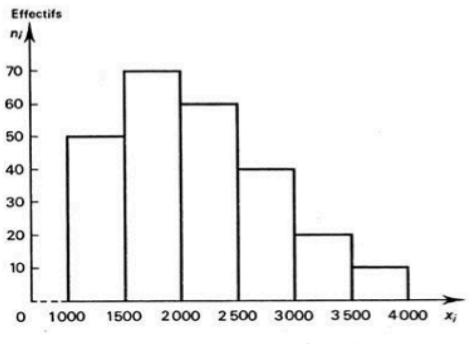
Exemple : soit la série suivante qui représente le nombre de foyers selon le nombre de personnes par foyer (tableau). Le diagramme à gauche représente graphiquement ce tableau.



Nombre d'enfants par foyer « x_i »	Nombre de foyer concernés f_i
0	12
1	31
2	29
3	11
4	6
5	3

- *Histogrammes* : Ils s'adaptent aux cas d'une variable continue quantitative dont les valeurs peuvent être classées en intervalles. L'axe des abscisses représente les classes et l'axe des ordonnées représente les valeurs sous forme de rectangles.

Exemple : soit la série suivante qui représente le nombre de personne selon la tranche de salaire (tableau). Le diagramme à gauche représente graphiquement ce tableau.

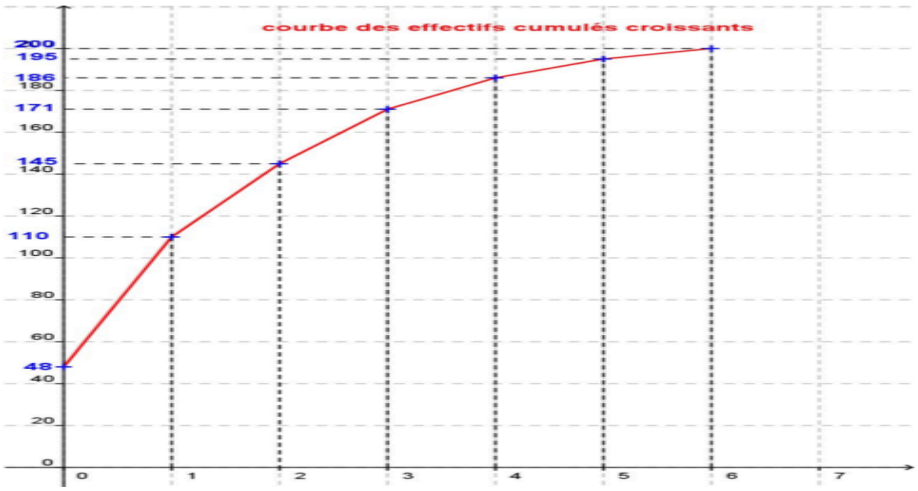


Salaire en € : x_i	Effectifs n_i
1000 à 1500	50
1501 à 2000	70
2001 à 2500	60
2501 à 3000	40
3001 à 3500	20
3501 à 4000	10
	250

- *Graphiques cumulatifs* : ils permettent de représenter les cumuls d'effectifs d'une série ordonnée. Si les effectifs initiaux ne correspondent pas à des cumuls, on les calcule d'abord avant de représenter le diagramme.

Exemple : le tableau suivant représente l'évolution de salaire selon le grade. La deuxième ligne représente le montant d'évolution et la troisième ligne représente le cumul de salaire.

Modalités	0	1	2	3	4	5	6
Effectifs	48	62	35	26	15	9	5
ECC	48	110	145	171	186	195	200

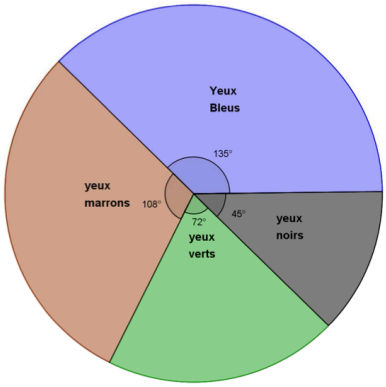


- *Diagrammes en secteur* : ce diagramme permet de représenter des proportions d'effectifs par rapport à la totalité. Avant de le dessiner, on calcule la proportion de chaque valeur de variable (individu). On calcule ensuite l'angle de chaque valeur de variable.

Exemple : soit le tableau suivant qui donne les effectifs et fréquences des couleurs d'objets observés. Les modalités sont les couleurs bleu, marron, vert et noir. Les fréquences sont traduites en angles en les multipliant par 3,6.

Modalités	bleu	marron	vert	noir	total
Effectifs	15	12	8	5	40
Fréquences	0,375	0,3	0,2	0,125	1
Fréquences en %	37,5	30	20	12,5	100
Angle (en °)	**135**	**108**	**72**	**45**	360

x 3,6



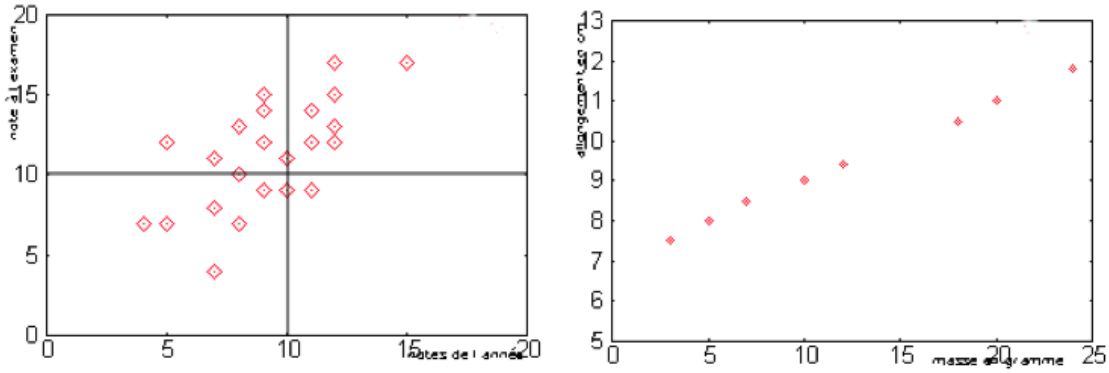
2.2. Cas de deux variables : il arrive souvent qu'on ait besoin d'analyser deux variables à la fois et qu'on cherche la relation entre elles. Par exemple : la relation entre la taille des enfants et leur âges. On note ce deux variables x et y.

- Notions : parmi les notions qui font intervenir les deux variables à la fois sont
  - La covariance : $$\mathrm{cov}(x, y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$$

- Le coefficient de corrélation linéaire: $$r = \frac{cov(x,y)}{\sigma_x \sigma_y}$$

La covariance décrit la relation entre les changements des deux variables. Si elle est positive, aux grands écarts d'une variable correspondent de grands écarts de la deuxième et vice-versa. Par contre, une covariance négative signifie qu'aux grands écarts d'une variable correspondent de petits écarts de la deuxième.

- Représentation graphique : chaque individu est représenté par un point dans un plan. Les valeurs d'une variable sont placées sur un axe et les valeurs de la deuxième variable sur l'autre axe. Cette représentation est appelée nuage de points. Elle permet de déduire visuellement s'il y a une relation entre les valeurs des deux variables. Dans les graphiques ci-dessous, le nuage de points à gauche témoigne que les valeurs des d'une variable sont indépendantes des valeurs de l'autre. Le nuage à droite montre qu'il se peut qu'il y ait une relation entre les valeurs.



Lorsqu'il peut exister une relation entre les valeurs des deux variables, on procède à l'ajustement

2. 3. Cas multidimensionnel : le cas multidimensionnel concerne plusieurs variables à la fois. La représentation graphique n'est pas adaptée à l'être humaine. Le traitement de ces valeurs fait intervenir les méthodes d'analyse de données.

3. Analyse de données

- *Définition* : l'analyse de données regroupe une famille de méthodes pour décrire un grand nombre de données avec comme objectif de faire ressortir les relations entre elles ou de comprendre ce qui les rend homogènes.

- *Exemples de méthodes*
  - Analyse en composantes principales (ACP) : réduire *p* variables corrélées en *q* variables non corrélées.
  - Analyse factorielle des correspondances (AFC) : trouver des liens ou correspondances entre deux variables qualitatives (nominales) dans des tableaux de contingence.
  - Analyse des correspondances multiples (ACM) : L'ACM est l'équivalent de l'ACP pour les variables qualitatives et elle se réduit à l'AFC lorsque le nombre de variables qualitatives est égal à 2.

Chapitre 3 : Le Clustering (classification automatique)

1. Introduction

- ***Définition*** **:** technique visant à grouper des objets dans des classes a priori inconnues de sorte à ce que les objets d'une même classe soit similaires et les classes ne soit pas similaires.
- ***Objectifs*** **:** (1) Abstraire les objets en classes (conceptualisation). (2) utiliser les classes obtenues à des fins diverses : (i) marketing (ciblage de clients, profiling), (ii) navigation Web (personnalisation du contenu par classe d'utilisateurs), (iii) Détection de communautés, etc.

2. Recherche de clusters

- On dispose d'un ensemble d'objets (individus, voitures, etc.) qu'on veut regrouper.
- Les objets ne sont pas classés a priori.
- Tous les objets sont décrits par un ensemble de caractéristiques (variables)
- Les variables sont de divers types mais ne sont pas toutes adéquates à la classification.
- Chaque objet peut être considéré comme un vecteur dans un espace multidimensionnel. Les objets similaires peuvent être proches géométriquement.

3. Algorithme K-means (K-moyennes)

Il existe beaucoup d'algorithmes de classification automatique, dont le plus ancien est celui des K-means (k-moyennes) avec beaucoup de variations pour elle.

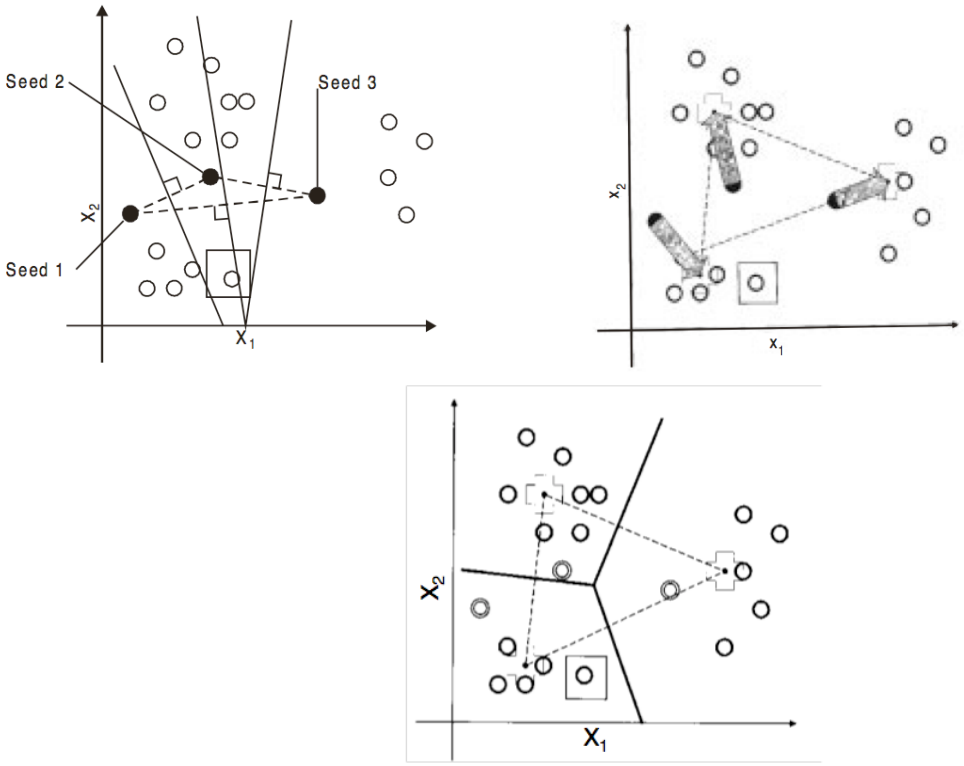
a. ***Principe de base :*** le principe de base de K-means est regrouper les objets en se basant à leur distance par rapport à un centre (centroïd). Initialement, les centres sont choisis aléatoirement, mes ils seront mis à jour par re-calcul. Les cluster initiaux seront affinés, voire changés. La classification finale correspondra à une stabilisation des clusters : aucun objet ne change d'appartenance à son cluster.

b. ***Description de l'algorithme :*** Soit les N objets à classifier (pour les besoins d'analogie géométrique, on désigne les objets comme étant des points)

- Choisir K centres aléatoires (seed).
- Calculer la distance entre chacun des points restants et chaque seed.
- Calculer le centre de chaque cluster. La valeur de chaque caractéristique de chaque centre est égale à la moyenne des valeurs de la caractéristique pour tous les points du cluster correspondant.
- Réaffecter les points selon leur distance par rapport au centre de chaque cluster : de nouveaux cluster peuvent être générés.
- Recalculer le centre C de chaque cluster
- S'arrêter lorsqu'aucun point ne change de cluster (stabilisation des clusters).

Illustration : On suppose que les objets sont décrits par deux caractéristiques X1 et X2. Les trois

figures suivantes (de Nagabhushana) montrent les premières itérations de k-means. La figure à gauche montre le choix des seeds initiaux (cercles noirs) et la formation des premiers clusters (trois. La figure à droite montre les centroids (symbole +). La troisième figure montre les nouveaux clusters. Le point entouré d'un carré est le point ayant changé de cluster après le calcul des centres.



4. ***Exemple illustratif*** (ref. Kardi) : soit les quatre objets décrits par deux caractéristiques avec les valeurs suivantes : $X_1(1, 1)$, $X_2(2, 1)$, $X_3(4, 3)$, $X_4(5, 4)$ (voir figure 1). On choisit les X_1 et X_2 comme seeds (étoiles) et on les désigne par C_1 et C_2. On calcule la distance entre chaque point et chaque centroid. On utilise la distance euclidienne (voir plus loin). On obtient la matrice suivante :

X_1	X_2	X_3	X_4	
0	1	3,61	5	C_1
1	0	2,83	4,24	C_2

On remarque que les points X_2, X_3 et X_4 sont proches à C_2 et X_1 à C_1. On obtient deux cluster composés de (X_1) et (X_2, X_3, X_4). Le calcul des nouveaux centres donne $C'_1(1,1)$ et $C'_2(11/3, 8/3)$. On recalcule les nouvelles distances. On obtient le tableau suivant.

X_1	X_2	X_3	X_4	
0	1	3,61	5	C'_1
3,14	2,36	0,47	1,89	C'_2

Selon cette matrice, on déplace X_2 au cluster à gauche et on obtient les clusters de la figure 3. On recalcule les centres (étoiles de la figure 3). On trouve les centres $C''_1(1.5, 1)$ et $C''_2(4.5, 3.5)$. On calcule les distances et on obtient les centres suivants

X_1	X_2	X_3	X_4	
0,5	0,5	3,20	4,61	C''_1
4,30	3,54	0,71	0,71	C''_2

La matrice montre qu'aucun point ne change de cluster. On arrête les itérations.

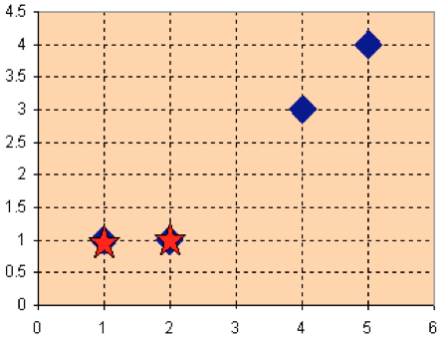


Figure 1

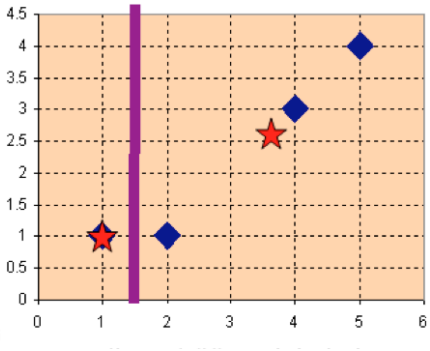


Figure 2

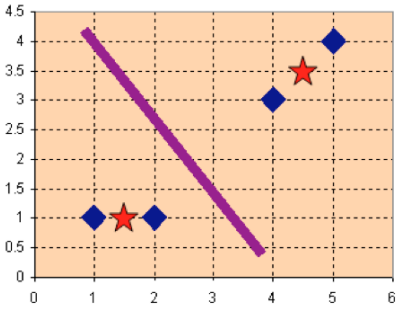


Figure 3

5. **Types de variables pour le clustering :** les types de données pour le clustering ne sont pas tous adéquats. On les classe par ordre croissant d'adéquation comme suit :

   - Variables symboliques (couleurs, noms, etc.)
   - Rangs (1, 2, 3, etc.)
   - Les intervalles (degrés de températures, etc.)
   - Les vraies mesures (âge, longueurs, volumes, etc.)

Les intervalles et les vraies mesures sont les plus adéquats au clustering. Les variables symboliques et les rangs doivent être transformées pour être utilisées. Cependant, les transformations ne donnent pas toujours des intervalles ou mesures ayant un sens et comparables.

6. **Mesures de similarité :** pour trouver les objets associés, on utilise différentes mesures de similarité.
   a. *Distance* : le calcul de distance dépend de la nature des données

- Intervalles et vraies mesures : tout d'abord, nous standardisons les données. On calcule l'écart absolu moyen par la formule $s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + ... + |x_{nf} - m_f|)$ où m représente la moyenne des valeurs. Ensuite, on divise l'écart entre chaque mesure et sa moyenne par l'écart-type moyen. $z_{if} = \frac{x_{if} - m_f}{s_f}$

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	2

Une fois les mesures standardisées, nous calculons les distances. Les formules sont

- Formule de Minkowski $d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + ... + |x_{i_p} - x_{j_p}|^q)}$
- q = 1 : distance de Manhattan : $d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + ... + |x_{i_p} - x_{j_p}|$
- q = 2 : distance euclidienne : $d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2)}$

- Variables binaires : on élabore une table de contingence (voir exemple en cours) pour calculer quatre valeurs, désignées par a, b, c et d. « a » désigne le nombre de cas où l'objet i possède un 1 et l'objet i possède un 1. Pour b, c et d, on cherche les respectivement le nombre des cas où on a (1, 0), (0, 1), (0, 0).

  - Une fois les valeurs de a, b, c et d trouvées, on aura deux mesures :
    - Coefficient d'appariement simple : $d(i,j) = \frac{b+c}{a+b+c+d}$
    - Coefficient de Jaccard : $d(i,j) = \frac{b+c}{a+b+c}$

Remarque : lors du codage des variables booléennes en binaire, l'assignation des valeurs 1 et 0 dépendra de la symétrie des objets.

- Si les objets sont symétriques (pas de valeur booléenne fréquente), on code au hasard les deux valeurs booléennes
- Si les objets sont asymétriques (ex : test de virus), on code par 1 la valeur la moins fréquente.

La distance sera calculée en utilisant les variables asymétriques.

b. *Angle entre les vecteurs :* ce calcul ne tient pas compte des grandeurs des mesures.
c. *Nombre de caractéristiques communes* : on calcul le nombre d'appariements

$$d(i,j) = \frac{p-m}{p}$$

où *m* est égal au nombre de caractéristiques communes et *p* est le nombre total de caractéristiques.

7. **Choix du nombre K** : il n'existe pas de meilleurs K ou de mauvais K pour le clustering, puisque ce nombre est fourni par l'utilisateur pour spécifier le nombre de clusters souhaités. Cependant, les clusters obtenus peuvent être évalués. Les bons clusters seront le résultat d'un bon choix de *k* et inversement.

8. **Classification hiérarchique ascendante (AHC) :** appelée *agglomerative clustering*, dans ces méthodes, la construction des clusters se fait à partir des objets individuelles en les regroupant selon leur similarité. Les clusters formés au départ seront petits mais au fur et à mesures, des clusters plus grands seront formés jusqu'à aboutir au clusters qui contient tous les objets. Le résultat de la AHC n'est pas un seul mais plusieurs niveaux de clusters représentés par un dendrogramme (voir figure ci-dessous). Le choix du niveau dépend de l'application souhaitée. Les étapes sont les suivantes :

   - Création d'une matrice de similarité entre chaque pair d'objets. On utilise les distances telles que décrites plus haut. La matrice de similarité se présente comme suit :

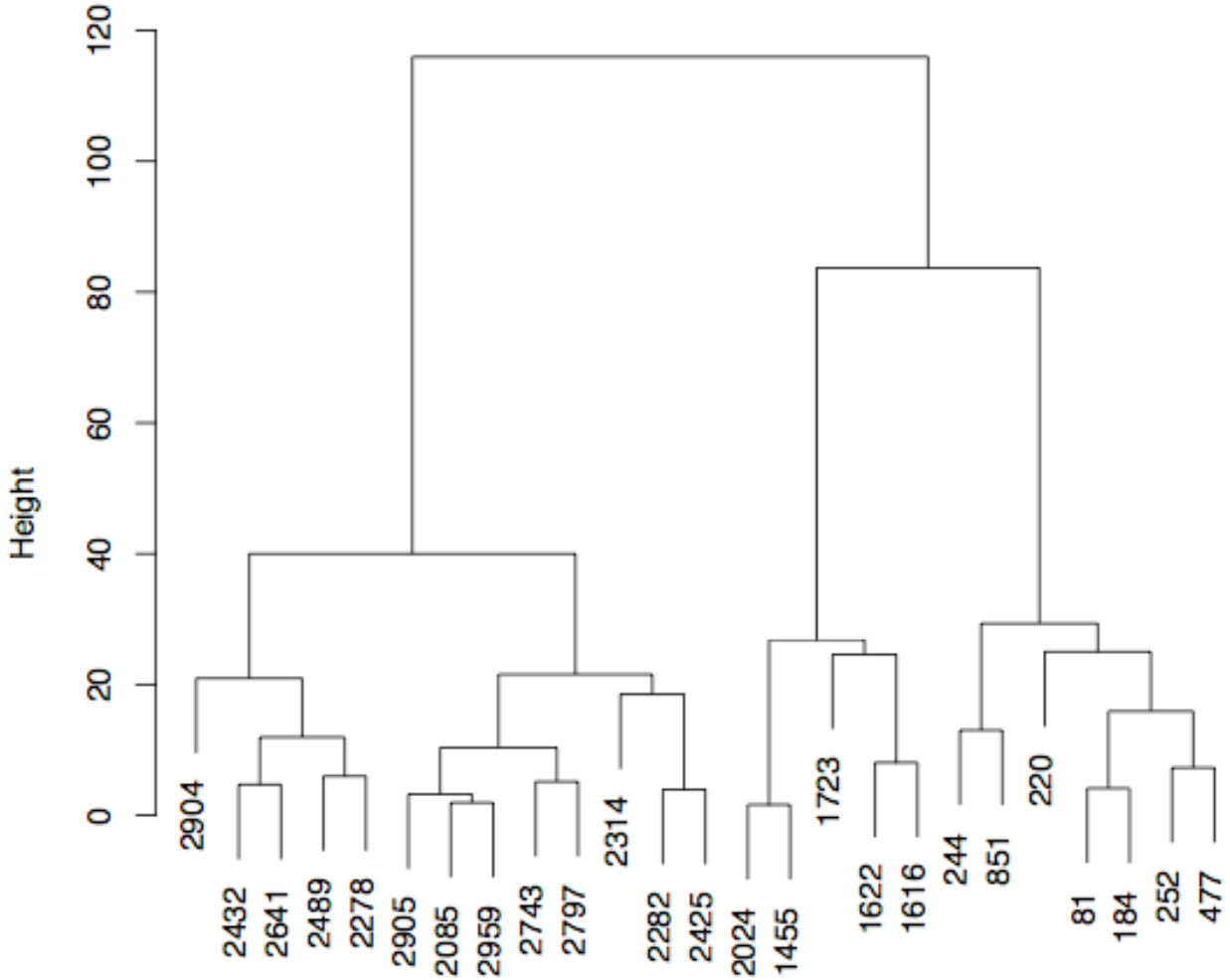
$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \ldots & \ldots & 0 \end{bmatrix}$$

   on considèe chaque objet comme cluster.

   - Chercher la plus petite valeur de la matrice. Les clusters seront fusionnés en un seul. On met à jour la matrice de similarité en remplaçant les deux lignes des clusters fusionnés par une seule ligne et en mettant à jour les distances.
   - On répète les étapes précédentes jusqu'à l'obtention du cluster contenant tous les objets.

Mesure de distance entre clusters : hormis la première étape, les clusters obtenus ne correspondent pas à des objets individuels ; les distances présentées précédemment ne sont pas adéquates. Pour trouver la distance entre deux clusters, on utilise une des approches suivantes :

- la distance entre clusters est égale à la distance entre les objets les plus proches, chacun d'un cluster (linkage simple).
- la distance entre clusters est égale à la distance entre les objets les plus éloignés, chacun d'un cluster (linkage complet).
- la distance entre clusters est égale à la distance entre les centroïds (comparaison entre centroïds).


Height
0
20
40
60
80
100
120
2904
2432
2641
2489
2278
2905
2085
2959
2743
2797
2314
2282
2425
2024
1455
1723
1622
1616
244
851
220
81
184
252
477

Chapitre 4 : Les règles d'association

1. **Introduction :** certaines données transactionnelles renferment des connaissances sous forme d'associations entre divers objets de différents types mais appartenant aux mêmes transactions. Par exemple, le panier des achats peut contenir des articles de types différents (lait, pain, beurre). L'analyse et la découverte des associations de types « si un client achète du lait alors il achète du pain » ou la découverte des articles associés peut être utile à différentes fins, telles que la réorganisation des rayons, les promotions, etc. Cependant, deux problèmes se présentent lors de la recherche de ces associations

    a. Le volume de données peut rendre la découverte d'association fastidieuse.
    b. Certaines associations peuvent se produire par hasards.

2. **Exemple Illustratif** : supposons que les achats concernent les articles suivants : pain, lait, jus, beurre, œufs, cola. On fait abstraction des quantités. Une transaction est matérialisée par un ticket de caisse. Chaque achat peut être vu comme un objet et chaque article comme une variable. Dans ce cas, les variables sont binaires car ils peuvent avoir la valeur 1 si l'article est acheté ou 0 sinon. On peut mettre en forme les achats comme suit.

ID du ticket	Pain	Lait	Jus	Beurre	Œufs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	0

Dans le tableau ci-haut, on désigne par

- **Ensemble d'articles (*itemset*) :** l'ensemble d'articles qui sont achetés ensemble. Ex : dans la transaction 1, le itemset contient du pain et du lait. Deux transactions ou plus peuvent être identiques.

- **Règle d'association :** l'implication *Si A alors B* où A et B désigne deux sous-ensembles disjoints d'articles de l'ensemble total.

3. **Mesures d'évaluation des règles d'association :** la découverte des associations peut résulter en plusieurs combinaisons d'articles qui ne sont pas toutes utiles car s'agissant d'associations rares ou hasardeuses. Pour garder les bonnes associations, on utilise deux critères d'évaluation. Soit A et B deux ensembles d'articles :

    - *Support d'une règle d'association* : il s'agit du rapport entre le nombre de transactions qui contiennent A et B sur le nombre total de transactions.
    - *Confiance accordée à une règle*: il s'agit du rapport entre le nombre de transactions qui contient A et B sur le nombre de transactions qui contiennent A.

→ Une bonne règle est celle ayant un support et une confiance élevés.

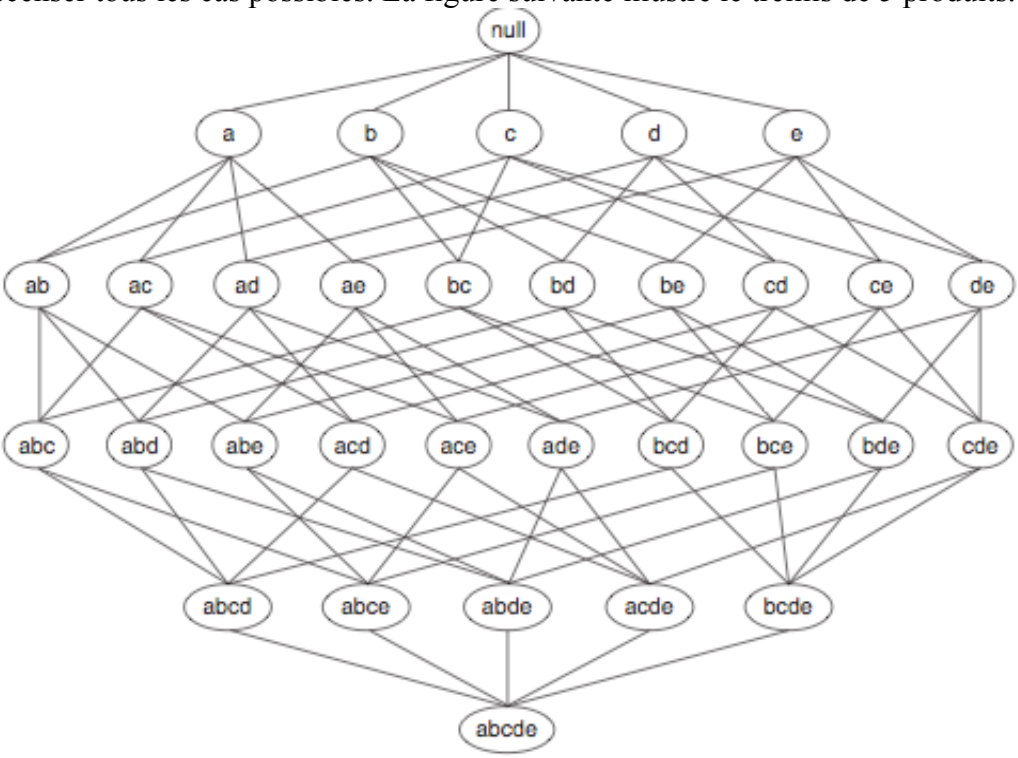
Exemple : soit la règle si achat du Pain alors achat de Lait.

- Support : 3 (nombre de transactions contenant le pain et le lait) /5= 60%
- Confiance : 3 (nombre de transactions contenant le pain et le lait) / 4 (nombre de transaction contenant du pain) = 75%.

4. Démarche de d'extraction des règles d'association

a. ***Bien choisir les item et les niveaux :*** les items à choisir correspondent à des produits dans une supérette ou à autre chose dans d'autres domaines. Dans le cas des produits par exemple, il faut sélectionner les produits (pain, beurre, viande) ou leurs catégories (produits laitiers, vêtements, etc.), comme on peut mélanger différents niveaux hiérarchiques des produits selon le besoin d'analyse.
b. ***Fixer un degré d'exigence sur les règles :*** lorsque le volume de données est suffisamment grand, le nombre de règles peut également être grand et leur calcul fastidieux → il faut fixer les deux paramètres **support** et **confiance** pour aboutir à des règles de qualité d'une part et limiter le nombre de règles produites d'autre part. Ce support et cette confiance sont **minimaux**.
c. ***Rechercher les itemsets fréquents :*** ils correspondent aux itemsets ayant un support supérieur ou égal au support minimal.
d. ***Produire les règles d'associations :*** à partir des itemsets fréquents, on garde les règles ayant une confiance supérieure ou égale à la confiance minimale.

5. **Génération des itemsets fréquents :** on peut se baser sur un treillis d'itemsets pour recenser tous les cas possibles. La figure suivante illustre le treillis de 5 produits.



La génération des itemsets fréquents se fait par le test de présence de chaque itemset dans les

transactions, ce qui peut devenir complexe devant un grand nombre de transactions. Pour réduire la complexité, on procède à l'élagage du treillis en se basant sur le support.

- ***Elagage par support (support-based pruning)*** : il consiste à réduire le treillis en se basant sur le principe suivant : *Si un itemset est fréquent (support supérieur ou égal au support minimum) alors tous ses sous-itemsets sont aussi fréquents*. Inversement, *si un itemset n'est pas fréquent, tous ses super-itemsets ne sont pas fréquents*.

- ***Algorithme Apriori*** : il s'agit de l'un des premiers algorithmes de recherche des règles d'associations. L'algorithme commence par l'énumération des itemsets de cardinalité 1. Les itemsets fréquents sont ceux ayant un support égal ou supérieur au support minimum. A partir de ces itemsets fréquents, on génère les itemsets de cardinalté 2 et on ne garde que les fréquents. On procède ainsi jusqu'à ce que l'on ne puisse plus générer d'itemsets. La figure suivante (Tan et al., 2006) représente le pseudo code de l'algorithme Apriori

1: $k = 1$.
2: $F_k = \{ i \mid i \in I \wedge \sigma(\{i\}) \geq N \times minsup\}$. {Find all frequent 1-itemsets}
3: **repeat**
4: $k = k + 1$.
5: $C_k = \text{apriori-gen}(F_{k-1})$. {Generate candidate itemsets}
6: **for** each transaction $t \in T$ **do**
7: $C_t = \text{subset}(C_k, t)$. {Identify all candidates that belong to t}
8: **for** each candidate itemset $c \in C_t$ **do**
9: $\sigma(c) = \sigma(c) + 1$. {Increment support count}
10: **end for**
11: **end for**
12: $F_k = \{ c \mid c \in C_k \wedge \sigma(c) \geq N \times minsup\}$. {Extract the frequent k-itemsets}
13: **until** $F_k = \emptyset$
14: Result $= \bigcup F_k$.

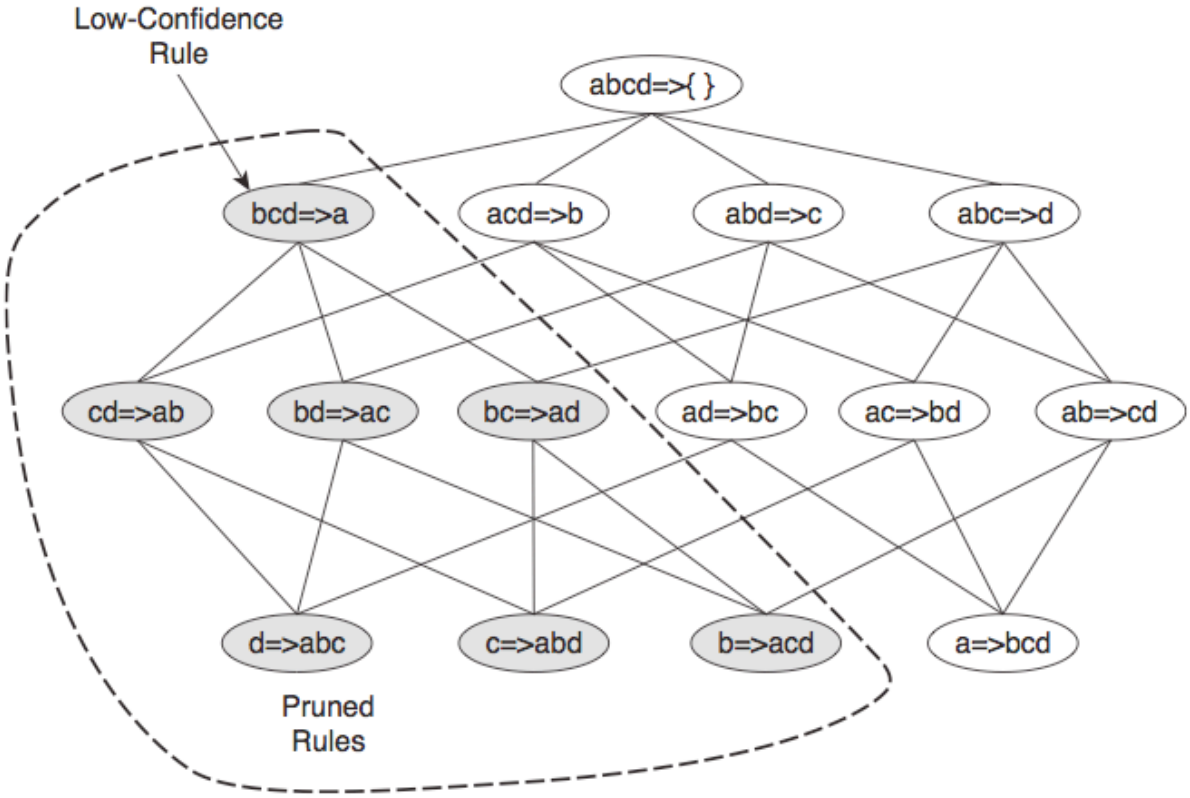
où F_K désigne un itemset de cardinalité K et C_K désigne les itemsets candidats avant le prunning, *T* désigne l'ensemble de transactions et *t* désigne une transation.

- Génération des itemsets candidats : il existe différentes manières de générer les itemsets fréquents (étape 5 de l'algorithme). La procédure apriori-gen(F_{K-1}) pour générer les itemsets fréquents candidats (C_K) de cardinalité K à partir des itemsets fréquents de cardinalité K-1 est basée sur la fusion des ensembles F_{K-1} ayant les K-2 premier articles identiques et le (K-1)^{ème} article différent. Par exemple, on fusionne les ensembles (a, b, c) et (a, b, d) en (a, b, c, d) mais pas (a, b, c) et (b, d, e).

6. **Génération des règles :** la génération des règles d'association se fait à partir des ensembles d'items fréquents. Le nombre de règles candidates qu'on peut générer est de 2^K-2 en ignorant les règles avec antécédent ou conséquence nulles. Comme la génération et l'exploration de toutes les règles peuvent également devenir coûteuses, on procède à l'élagage des règles au niveau du treillis. Pour effectuer cet élagage, on se base sur le théorème suivant.

Théorème : soit Y un itemset fréquent et X un sous-ensemble de Y. Soit X' un sous-ensemble de X. alors *Si la règles X → Y-X ne satisfait pas le seuil de confiance, alors toute règle X' → Y-X ne peut pas satisfaire le seuil de confiance également.*

- ***Génération des règles dans l'algorithme Apriori :*** la figure suivante (Tan et al. 2006) le treillis généré à partir d'un itemset de cardinalité 4. Les règles en gris représentent les règles élaguées à cause d'une confiance baisse de la règle bcd → a.



La génération des règles se fait de manière incrémentale selon le pseudo algorithme suivant :

1) On prend en entrée chaque itemset fréquent (par exemple, l'ensemble contenant abcd)
   a. A partir de la règle abcd → **Φ**, on génère toutes les règles possibles.
2) Pour chaque règle générée, on teste sa confiance
3) si elle est inférieure au seuil minimal, la règle est éliminée et aucune génération ne se fait à partir d'elle.
4) Sinon, la règle est maintenue.
5) Reboucler sur l'étape 2.

Support de cours en Fouille de données
Chapitre 5 : Les arbres de décision (classification)

1. **Introduction :** les arbres de décision sont un moyen qui permet de séparer des individus dans des groupes selon des règles ou de prévoir la valeur d'une variable continue (cible) à partir de variables en entrée. Il s'agit d'apprentissage supervisé car les groupes (classes) ou la variable à prévoir sont prédéfinis. On cherche alors à induire un ensemble de règles à appliquer à un nouvel objet afin de déterminer sa classe d'appartenance ou à connaitre la valeur de la variable. L'appellation *arbre* provient du fait que les règles s'enchainent de sorte que chaque règle correspond à un test donnant lieu à un nœud et les alternatives de réponse au test donnent lieu aux branches. Lorsqu'un arbre de décision est utilisé pour prédire une variable qualitative (i.e. la classe d'appartenance d'un objet), on parle *d'arbre de classification*. Par contre, lorsqu'il est utilisé pour prédire des variables continues, on parle *d'arbres de régression*. Dans ce cours, nous nous intéressons au premier cas.

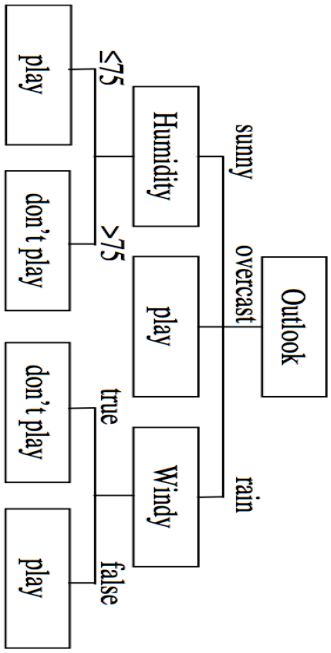
2. **Exemples d'utilisation :** les utilisations des arbres de classifications sont multiples, e.g.

    a. Un client X est-il à haut risque pour un prêt ? (classes : haut / moyen/ faible risque)
    b. Le risque qu'une population attrape une maladie
    c. Un objet détecté par un radar (véhicule, individu, immeuble, arbre).
    d. Le degré de ressemblance de personnes à un criminel recherché (léger, fort…).

3. **Exemple illustratif :** l'exemple repris par beaucoup d'auteurs est celui du joueur de golf qui décide de jouer ou pas sur la base du temps qu'il fait. Le tableau repris dans ce cas est illustré ci-dessous à gauche. Dans cet exemple, chaque nouveau jour représente un individu sur lequel on veut décider comme jour de jeu ou de non jeu (classes *jouer* et *ne pas jouer)*. Les autres variables sont des variables en entrée. La figure à droite illustre un arbre de classification qu'on peut facilement interpréter. On remarque que selon les données qui ont servi à induire l'arbre, la variable température n'a pas été utilisée.

4. **Formulation du problème :** les données nécessaires à l'induction de règles de décision sont appelées ensemble d'apprentissage (training set). Elles se présentent sous une forme tabulaire *individu*/*attribut* et peuvent être qualitatives (e.g. Outlook, Windy, Class) ou numériques (Temp et Humidity). On distingue une variable pour être la variable à prédire. Dans l'exemple ci-dessous, il s'agit de la variable Class avec deux modalités : *play* et *don't play*. Le problème consiste à élaborer un arbre où chaque nœud (appelé nœud de décision) consiste en un test sur la valeur d'un attribut. Les feuilles de l'arbre correspondent aux valeurs de l'attribut de prédiction. Les tests de valeur d'attributs diffèrent selon la nature des attributs (qualitatifs ou numériques).

Outlook	Temp (°F)	Humidity (%)	Windy	Class
sunny	75	70	true	play
sunny	80	90	true	don't play
sunny	85	85	false	don't play
sunny	72	95	false	don't play
sunny	69	70	false	play
overcast	72	90	true	play
overcast	83	78	false	play
overcast	64	65	true	play
overcast	81	75	false	play
rain	71	80	true	don't play
rain	65	70	true	don't play
rain	75	80	false	play
rain	68	80	false	play
rain	70	96	false	play



5. **Algorithme de base pour la construction d'un arbre de décision :** avant de présenter l'algorithme, les données d'apprentissage doivent satisfaire les conditions suivantes (i) avoir une forme tabulaire individu/attribut, (ii) la classe cible doit être prédéfinie (iii) des données suffisantes (centaine, millier d'individus ou plus).

   a. **Principe de l'algorithme :** l'algorithme est récursif. A partir d'un nœud initial, on choisit à chaque étape un attribut pour la division de l'ensemble d'apprentissage en sous-ensembles : chaque sous-ensemble correspond à l'attribut sélectionné et donne lieu à un nœud fils. Chaque branche du père vers le fils correspond à une valeur de l'attribut sélectionné. L'algorithme s'arrête lorsqu'il n'y a plus d'attributs de sélection.

   b. **Présentation de l'algorithme :** On présente l'algorithme CLS de Hoveland et Hunt (1950). Dans ce cas, les valeurs de la classe de prédiction sont (+) et (-).
      i. A ← l'ensemble d'apprentissage E. Créer un nœud de A.
      ii. Si tous les exemples de E ont la valeur positive pour la classe de prédiction, alors créer un nœud P sous la racine et s'arrêter.
      iii. Si tous les exemples de E sont la valeur négative, alors créer un nœud N sous la racine et s'arrêter.
      iv. Sélectionner un attribut X ayant les modalités v_1, v_2, …v_N et partitionner E en N sous-ensembles E_i, chacun correspondant à une modalité. Pour chaque sous-ensemble, créer un nœud ayant comme libellé la modalité v_i.
      v. Pour chaque sous-ensemble E_i, E ← E_i, aller à ii.

6. **Problème de sélection des meilleurs (attributs) classifieurs :** l'algorithme précédent ne spécifie pas le critère de sélection des attributs à chaque étape. Ce choix peut être aléatoire, ou basé sur un critère particulier. Les arbres obtenus peuvent être différents en profondeur. Le choix du classifieur détermine l'efficacité de l'induction à partir de l'arbre de décision. On cite deux choix : l'utilisation de l'entropie et gain informationnel, et l'utilisation des fréquences.

7. **Sélection à base d'entropie / Algorithme ID3 :** l'entropie telles que définie par Shannon mesure la quantité d'information attendue lors de l'envoie d'un message. Elle est liée à l'incertitude par rapport à un phénomène donné. L'exemple courant et celui du jet d'une pièce de monnaie avec deux valeurs possible (pile ou face) : l'entropie est maximale lorsque la pièce n'est pas truquée, i.e. les deux faces ont la même probabilité. L'entropie est nulle lorsqu'on possède une certitude (si la pièce contient deux faces *piles*). L'entropie est utilisée dans l'estimation du gain information (ou réduction d'incertitude) par rapport à une valeur de la classe à prédire. L'attribut choisi à une étape est celui qui minimise l'incertitude par rapport aux valeurs de la classe à prédire. La réduction d'incertitude (d'entropie) est également appelée le gain informationnel.

   a. **Entropie de Shannon :** soit un ensemble de mots à coder m_1, m_2, …sur un canal binaire C, ayant des probabilités d'apparition respective p_1, p_2, …. L'entropie est donnée par la formule

   $$\text{Entropie(C)} = \sum_{i=1}^{c} - p_i \log_2 p_i$$

   b. **Analogie avec les arbres de décision :** par rapport aux arbres de décision, chaque valeur d'un attribut donné correspond à un mot dans l'entropie de Shannon. La probabilité de chaque valeur est calculée par rapport à un sous-ensemble de données courant, i.e. celui obtenu par la division des donnée.

   c. **Utilisation de l'entropie / gain informationnel pour la sélection du meilleur classifieur :** pour décider quel attribut choisir comme attribut de décision à une étape donnée, on calcul pour chaque attribut le gain informationnel. Soit S l'ensemble courant de données et soit A un attribut. Le gain en information induit par le choix de A comme attribut de décision est donné par la formule

$$\text{Gain } (S, A) = \text{Entropy } (S) - \sum_{v \in \text{ Values } (A)} \frac{|S_v|}{|S|} \text{ Entropy } (S_v)$$

Où Entropy(S) est l'entropie de l'ensemble de données courant, $|S_v|$ est le nombre d'exemples ayant la valeur *v* dans l'ensemble S et |S| désigne la cardinalité de l'ensemble S. L'attribut choisi est celui qui correspond au plus grand gain informationnel.

Exemple illustratif : on reprend l'exemple du joueur du golf. L'ensemble S initial contient 14 jours. Notons que les attributs *Temp* et *Humidity* sont continus et donc nous classons le premier dans 3 classes : hot (>= 80), mild (<=70) et cool (70 < temp < 80) et le second dans deux classes : normal (<85) et high (>=85). Nous condond aussi les valeurs de l'attribut Class en Yes ou Play et

No autrement. Nous obtenons le nouveau tableau suivant :

Outlook	Temp	Humidity	Windy	Class
Sunny	Cool	Normal	True	Y
Sunny	Hot	High	True	N
Sunny	Hot	High	False	N
Sunny	Cool	High	False	N
Sunny	Mild	Normal	False	Y
Overcast	Cool	High	True	Y
Overcast	Hot	Normal	False	Y
Overcast	Mild	Normal	True	Y
Overcast	Hot	Normal	False	Y
Rain	Cool	Normal	True	N
Rain	Mild	Normal	True	N
Rain	Cool	Normal	False	Y
Rain	Mild	Normal	False	Y
Rain	Mild	High	False	Y

Etape 1 : choix du premier attribut : on calcule le gain informationnel des choix Outlook, Temp, Humidity, Windy. Soir S l'ensemble initial. Valeurs (Outlook) : Sunny, Overcast Rain.

Gain (S,Outlook) = Entropie (S) -SOMME $_{v \text{ dans } \{sunny, overcast, rain\}}(|S_v|/|S| * \text{Entropie}(S_v))$
= Entropie (S) - (5/14)*entropie (S_{sunny}) – (4/14)*entropie($S_{overcast}$) – (5/14)*entropie(S_{rain})= 0.94 - (5/14)*0,97-0-(4/14)*0,97= **0.31.**
Par la même manière, on trouve Gain (S, Temp) = **0.04** , Gain(S, Humidity)=**0.005** et Gain(S, Windy) = **0.05** → on choisit Outlook comme premier attribut de partitionnement. La racine de l'arbre est l'attribut Outlook avec trois valeurs : *Sunny*, *Overcast* et *Rain*. En plus, on aura les trois tableaux suivants

Temp	Humidity	Windy	Class
Cool	Normal	True	Y
Hot	High	True	N
Hot	High	False	N
Cool	High	False	N
Mild	Normal	False	Y

Outlook *Sunny*

emp	Humidity	Windy	Class
Cool	High	True	Y
Hot	Normal	False	Y
Mild	Normal	True	Y
Hot	Normal	False	Y

Outlook *Overcast*

Temp	Humidity	Windy	Class
Cool	Normal	True	N
Mild	Normal	True	N
Cool	Normal	False	Y
Mild	Normal	False	Y
Mild	High	False	Y

Outlook *Rain*

On réitère l'algorithme pour chacun des sous-ensemble obtenus.

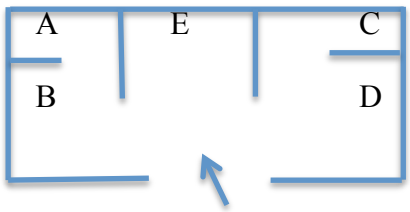
Série N°3 en Fouille de données (Règles d'association)

Exercice n°1 : Soit les données transactionnelles représentant les produits achetés dans un magasin. Une transaction correspond à un achat.

Transaction	A	B	C	D	E
1	1	1	0	0	0
2	1	0	1	1	1
3	0	1	1	1	0
4	1	1	1	0	1
5	1	0	1	1	1
6	1	1	1	1	1
7	0	1	0	0	1
8	1	0	1	1	1
9	1	1	0	1	1
10	0	1	1	0	0

1. Dessiner le treillis des itemsets de ces données.
2. En prenant comme support minimum la valeur 50%, donner les tableaux successifs des k-itemsets fréquents en illustrant les itemsets éliminés à chaque étape. Le passage entre deux tableaux consécutifs doit se faire par fusion des itemsets (algorithme Apriori).
3. En prenant comme confiance minimale la valeur 85%, générer les règles d'associations pour chaque itemset fréquent.

Exercice n°2 : on suppose que les produits de l'exercice 1 sont organisés dans le magasin comme illustré sur la figure avec les catégories suivantes C1 (A, B), C2(C, D) et C3 (E).



1. Expliquer comment utiliser les résultats des règles d'association (itemsets fréquents / règles d'association) pour augmenter les revenus du magasin.
2. Selon les résultats obtenus dans l'exercice précédent, proposer schématiquement une meilleure organisation des produits en utilisant les itemsets fréquents puis en utilisant les règles d'association.

Exercice n°3

1. Quelles sont les différentes combinaisons pour la recherche de règles d'association qu'on peut appliquer aux produits et leurs catégories de l'exercice 1.
2. En choisir une combinaison et donner son tableau de données transactionnelles.
3. Appliquer l'algorithme *Apriori* avec les mêmes paramètres de l'exercice 1. Comparer les résultats avec ceux de l'exercice 1.