Année universitaire	**2024-2025**		
Département	**Informatique**		
Filière	**Master IA & BioInfo**	**Année**	**M2**
Matière	**Techniques d'Apprentissage Automatique**		
Intitulé TD/TP :	**TD Méthodes ensemblistes**		

Exercice 1 : Apprentissage par arbre de décision

Une banque souhaite promouvoir une offre commerciale par courrier à ses clients. Pour cela elle fait appel à vous et à vos connaissances en fouille de données pour sélectionner ceux qui sont potentiellement intéressés. Quatre attributs descriptifs sont à votre disposition : l'âge (deux tranches : Jeune et Vieux), le sexe (H ou F) et le fait d'être propriétaire de son logement (O ou N) et le fait d'avoir fait des études supérieures (O ou N). L'attribut cible prend deux valeurs : O (intéressé) et N (pas intéressé). Le résultat d'une enquête préliminaire sur un panel représentatif de clients donne l'ensemble des données suivantes :

Table 1 : Base d'apprentissage

Client	Age	Sexe	Propriétaire	Etudes_supérieures	Intéressé
1	**J**	**H**	**N**	**O**	**N**
2	**J**	**F**	**N**	**O**	**N**
3	**J**	**H**	**O**	**N**	**O**
4	**J**	**H**	**O**	**N**	**O**
5	**J**	**H**	**N**	**O**	**O**
6	**V**	**F**	**O**	**N**	**N**
7	**V**	**H**	**O**	**O**	**O**
8	**V**	**F**	**O**	**N**	**N**
9	**V**	**H**	**O**	**N**	**N**
10	**V**	**F**	**O**	**O**	**N**

L'arbre de décision obtenu sur ces données est représenté ci-dessous.

Sexe=(H)
| Age=(J)
| | **Propriétaire**=(O): O
| | **Propriétaire**!=(O): N
| Age=(V): N
Sexe=(F)
| **Age**=(J): O
| **Age**=(V)
| | Etude_supérieure=(O): O
| | Etude_supérieure=(N): N

1. Calculer l'entropie globale.
2. Donner l'indice de Gini des deux nœuds en gras.

3. Evaluez l'erreur de cet arbre de décision final sur les données de test de la Table 2 en donnant la matrice de confusion et en calculant ensuite l'Accuracy ainsi que le rappel de la classe (Oui).

Table 2 : Base de test

Client	Age	Sexe	Propriétaire	Etudes_supérieures	Intéressé
11	**J**	**H**	**N**	**N**	**O**
12	**V**	**F**	**N**	**O**	**N**
13	**J**	**F**	**O**	**N**	**N**
14	**V**	**H**	**O**	**N**	**O**
15	**J**	**H**	**N**	**N**	**O**
16	**V**	**H**	**O**	**N**	**N**
17	**V**	**F**	**O**	**O**	**N**
18	**V**	**H**	**O**	**N**	**O**
19	**J**	**F**	**O**	**O**	**O**
20	**J**	**F**	**N**	**N**	**O**

Exercice 2 : Apprentissage ensembliste par Boosting

Le but de cet exercice est de construire un classifieur ensembliste par boosting de façon à classer les patients en 2 classes, malade *cardiaque* et *non cardiaque*, en fonction de divers attributs représentant les résultats de 5 examens médicaux. L'ensemble de ces données est représenté dans la table ci-dessous :

Table 1 : Base d'apprentissage

Patient	E1	E2	E3	E4	E5	Malade
1	0	0	0	1	0	Oui
2	0	0	1	1	0	Oui
3	0	0	0	0	0	Oui
4	0	0	1	1	0	Non
5	0	0	1	1	0	Non
6	0	0	0	0	0	Oui
7	1	0	0	0	1	Non
8	1	0	0	0	1	Non
9	1	0	0	0	1	Non
10	1	0	1	0	1	Oui
11	0	0	1	0	0	Oui
12	0	1	0	0	0	Oui
13	1	1	0	0	1	Oui
14	1	0	1	1	0	Oui
15	1	1	1	0	1	Non
16	0	0	0	0	0	Non
17	1	0	0	0	1	Non
18	0	0	1	1	0	Non
19	1	0	1	1	1	Oui
20	1	1	0	0	1	Oui
21	1	0	1	1	1	Oui
22	0	0	0	1	0	Non
23	0	0	0	0	0	Non
24	0	0	1	0	0	Non

L'idée du premier algorithme de boosting proposé par Schapire (1989) est d'utiliser un algorithme d'apprentissage qui peut être de natures très diverses (un arbre de décision, un réseau de neurones, etc.) sur trois sous-ensembles d'apprentissage.

i. On obtient d'abord une première hypothèse h_1 sur un sous-echantillon S_1 d'apprentissage de taille $m_1<m$ (m étant la taille de S l'échantillon d'apprentissage disponible qui est égale à 24 dans notre cas).

ii. On apprend alors une deuxième hypothèse h_2 sur un échantillon S_2 de taille m_2 choisi dans $S-S_1$ dont la moitié des exemples sont mal classés par h_1.

iii. On apprend finalement une troisième hypothèse h_3 sur m_3 exemples tirés dans $S-S_1-S_2$ pour lesquels h_1 et h_2 sont en désaccord.

iv. L'hypothèse finale est obtenue par un vote majoritaire des trois hypothèses apprises

1. Appliquez cet algorithme sur les données précédentes en utilisant pour l'apprentissage un arbre de décision de type CART (avec comme critère de segmentation le coefficient de Gini). Pour la première étape, utilisez les 10 premiers individus (m1 =10). Les arbres doivent être complet (sans élagage).
2. Evaluez les performances du modèle final obtenu sur les données d'apprentissage et les données de test de la Table 2 en donnant la matrice de confusion et en calculant ensuite le taux de réussite totale (accuracy), le rappel et la précision pour la classe *malade cardiaque*.

Table 2 : Base de test

Patient	E1	E2	E3	E4	E5	Malade
25	1	0	1	1	0	Oui
26	1	1	0	0	1	Oui
27	1	1	1	0	1	Oui
28	0	0	1	0	0	Non
29	0	0	0	0	0	Non
30	0	1	1	0	0	Oui

1. En appliquant un arbre de décision de type CART sur les 24 données d'apprentissage de la Table 1, nous obtenons l'arbre de décision ci-dessous. Evaluez les performances de cet arbre d'une manière identique à la question précédente. Que remarquez vous ? Conclure ?

```
E2=0
| E3=0
| | E1=1 : Non
| | E1=0 : Oui
| E3=1
| | E1=0
| | | E4=1: Non
| | | E4=0: Oui
| | E1=1: Oui
E2=1
| E3=1: Non
| E3=0: Oui
```

Exercice 3 : Problème

Dans le cadre d'un système de recommandation qui permet de suggérer à des clients de visiter les restaurants qui leurs conviennent, nous souhaitons savoir si un nouveau restaurant R_{p+1} qui vient d'être rajouter au catalogue est susceptible d'intéresser ou non un nouveau client C_{n+1} récemment arrivé. Proposer une solution issue du machine learning à ce problème tout en notant que les données sont décrites par les 3 matrices suivantes : la matrice Clients x Restaurants qui collecte pour un client donné s'il a visité ou non un restaurant donné, la matrice

caractéristiques de clients qui donne la description de chaque client et la matrice caractéristiques de restaurants qui donne la description de chaque restaurant. Nous avons la description du nouveau client C_{n+1} et du nouveau restaurant R_{p+1}.

Matrice Clients x Restaurants

Client\Restaurant	R_1	R_2	R_3	…	R_p
C_1	0	0	1		1
C_2	1	0	0		1
C_3	0	1	0		0
…					0
C_n	0	1	0		1

Matrice Caractéristiques des Clients

Client	Fc_1	Fc_2	Fc_3	…	Fc_m
C_1					
C_2					
C_3					
…					
C_n					
C_{n+1}					

Matrice Caractéristiques des Restaurants

Restaurant	Fr_1	Fr_2	Fr_3	…	Fr_k
R_1					
R_2					
R_3					
…					
R_p					
R_{p+1}					