Contrôle TD – Apprentissage

1h. **Documents non autorisés.**

Prenez soin de lire l'énoncé avant de commencer. La notation est donnée à titre indicatif.

Exercice :

RID	age	income	student	credit	C_i: buy
1	youth	high	no	fair	C_2: no
2	youth	high	no	excellent	C_2: no
3	middle-aged	high	no	fair	C_1: yes
4	senior	medium	no	fair	C_1: yes
5	senior	low	yes	fair	C_1: yes
6	senior	low	yes	excellent	C_2: no
7	middle-aged	low	yes	excellent	C_1: yes
8	youth	medium	no	fair	C_2: no
9	youth	low	yes	fair	C_1: yes
10	senior	medium	yes	fair	C_1: yes
11	youth	medium	yes	excellent	C_1: yes
12	middle-aged	medium	no	excellent	C_1: yes
13	middle-aged	high	yes	fair	C_1: yes
14	senior	medium	no	excellent	C_2: no

Soit le fichier ci-dessus possédant une variable de classe BUY, et découpé en 2 sous-ensembles: D_1 et D_2. **D_1 contient les 10 premiers objets**, et D_2 contient les **4 derniers (11 à 14)**.
D_1 sera *l'ensemble d'apprentissage* et D_2 sera *l'ensemble de validation*.

1- Traduire D_2 au format 'arff' de WEKA ; 1 pt
2- Combien y a-t-il d'attributs pertinents permettant de caractériser la classe BUY ; 0,5 pt
3- L'ensemble D_2 va être utilisé pour tester le modèle M_1 du **k-plus proche voisin** : k-PPV. Déterminer la classe des 4 objets de D_2; 3 pts
4- Donner la matrice de confusion de M_1 sur D_2 ; 0,5 pt
5- A partir de cette matrice de confusion, et après avoir **rappelé la formule de calcul** :
   a. Calculer le taux d'erreur apparente de M_1; 1 pt
   b. Calculer la précision de la classe C_1='yes'; 1 pt
   c. Calculer le rappel de la classe C_2='no' ; 1 pt
   d. Calculer la sensibilité de la classe C_1='yes'; 1 pt
   e. Calculer la spécificité de la classe C_2='no' ; 1 pt
   f. Calculer le taux de faux positifs de la classe C_1='yes'; 1 pt
   g. Calculer le taux de vrais positifs de la classe C_2='no' ; 1 pt
   h. Calculer la précision de M_1 ; 1 pt
   i. Calculer le rappel de M_1 ; 1 pt
6- On souhaite construire le modèle M_2 **d'arbre de décision** en utilisant *l'indice d'erreur en classification* (voir annexe).
   a. Construire l'arbre de décision M_2 sur l'ensemble d'apprentissage D_1 ; 3 pts
   b. Donner sa matrice de confusion sur D_2 ; 1 pt
7- Comment peut-on comparer les 2 modèles M_1 et M_2 ; 2 pts
8- **Bonus** : Proposer un algorithme de comparaison. 1 pt

ANNEXES

La ***précision*** pour une classe donnée mesure le taux d'exemples corrects parmi les exemples prédits dans cette classe.
Le ***rappel*** mesure le taux d'exemples corrects parmi les exemples de la classe.
Le taux de ***faux positifs*** d'une classe mesure le nombre d'objets positifs parmi ceux n'appartenant pas à la classe.
Le taux de ***vrais positifs*** d'une classe mesure le nombre d'objets positifs parmi les vrais objets de la classe.
Le taux de ***faux négatifs*** d'une classe mesure le nombre d'objets négatifs parmi ceux appartenant à la classe.
Le taux de ***vrais négatifs*** d'une classe mesure le nombre d'objets négatifs parmi ceux n'appartenant pas à la classe.
La ***sensibilité*** est la probabilité qu'un test soit positif si l'objet appartient à la classe.
La ***spécificité*** est la probabilité qu'un test soit négatif si l'objet n'appartient pas à la classe.

Arbres de décision

p(j / t) est la fréquence relative de la classe j au nœud t.

Indice de Gini pour le nœud t : $$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

Indice de Gini pour l'attribut *split* : $$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

Gain d'information avec l'indice de Gini pour l'attribut *split* : $Gain_{split} = Gini(r) - Gini_{split}$
Le nœud parent **r** a n objets, et est divisé en k partitions. La partition i possède n_i objets.

Entropie du nœud t: $$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

Gain d'information avec l'entropie pour l'attribut *split*: $$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$
Le nœud parent **p** a n objets, et est partitionné en k partitions. La partition i possède n_i objets.

Indice d'Erreur en classification au nœud t : $$Error(t) = 1 - \max_i P(i \mid t)$$

Gain d'information avec l'indice d'erreur en classification : $Gain_{split} = Error(r) - Error_{split}$
Le nœud parent **r** a n objets, et est partitionné en k partitions. La partition i possède n_i objets.

k-PPV: Proximité (Similarité, Dissimilarité), Distances

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

- Common situation is that objects, *p* and *q*, have only binary attributes
- Compute similarities using the following quantities
  M_{01} = the number of attributes where p was 0 and q was 1
  M_{10} = the number of attributes where p was 1 and q was 0
  M_{00} = the number of attributes where p was 0 and q was 0
  M_{11} = the number of attributes where p was 1 and q was 1
- Simple Matching and Jaccard Coefficients
  SMC = number of matches / number of attributes
  = $(M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

  J = number of 11 matches / number of not-both-zero attributes values
  = $(M_{11}) / (M_{01} + M_{10} + M_{11})$

Distance de Minkowski : $$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$
où r est un paramètre, n est le nombre de dimensions (attributs) et p_k et q_k sont, respectivement, les $k^{èmes}$ attributs (composants) des objets p et q.

r = 1 : City block (Manhattan, taxicab, L_1 norm) distance. Aussi appelée distance de Hamming pour vecteurs binaires.
r = 2 : Distance euclidienne