Examen Introduction aux Sciences des Données – Session 1

Durée 1h – Barême indicatif

Matériel autorisé : supports de cours + calculatrice non-programmable (téléphone interdit même pour sa calculatrice)

1 Arbre de décision (7 points)

Soit un problème de classification binaire sur des données décrites par trois variables (attributs $a_1 \in [1, 2, 3]$, $a_2 \in$ [V,F], et $a_3 \in [0, 1]$, étiquétées selon deux classes $\mathcal{Y} = \{+, -\}$. A partir d'un échantillon S de données étiquetées, on décide d'apprendre un arbre de décision $A(x)$ pour en dériver une séquence de règles de prédiction dans $\mathcal{Y}$ pour toute nouevlle donnée x. On sait qu'il y a 30 exemples de données dans S, 15 de la classe + et autant de la classe −.

1. Au début de la construction de l'arbre, on doit choisir un premier test sur les variables décrivant les données : chaque test t_j est à valeurs dans le domaine de la variable a_j, $j \in [1..3]$. La figure 1 indique comment les données se réaprtiraient en classes dans les noeuds créés par chacun des trois tests potentiels. On note $|S_{t,v}|$ le nombre d'exemples qui passent le test t avec la valeur v. Sous chaque feuille, les nombres d'exemples positifs et négatifs de S y arrivant sont indiqués (illustration : 9 exemples positifs et un exemple négatif parviennent à la première feuille en bas à gauche).

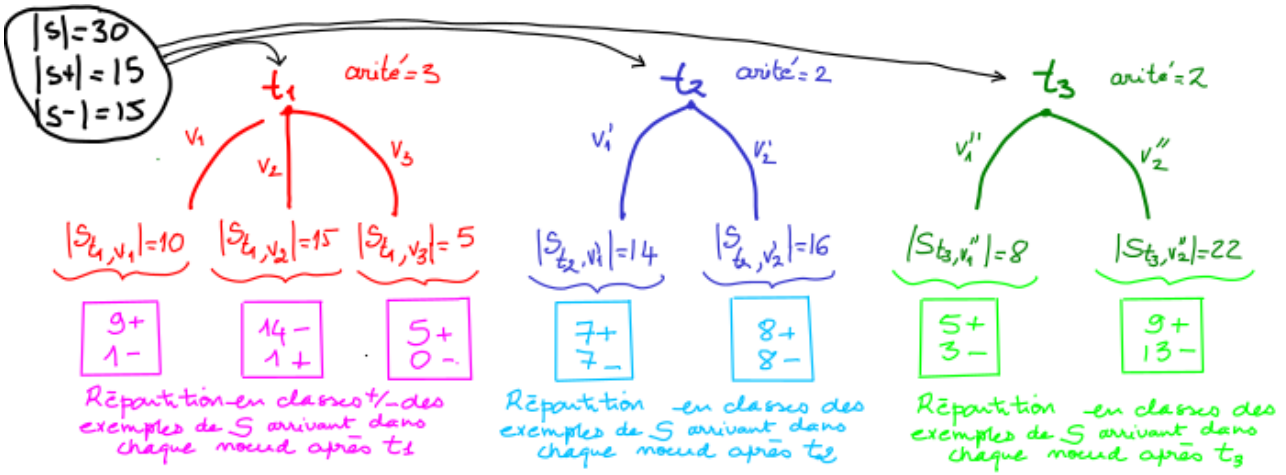

FIGURE 1 – Analyse pour le choix d'un test parmi 3

(a) Au vu des comptages indiqués sur la figure 1, quel est le meilleur test à choisir à la racine ? (justifier rapidement la réponse – sans faire de calcul, mais en terme de gain en information).

(b) L'arbre appris ne considère qu'un test à la racine (*un stump*), t_1, t_2 ou t_3 selon réponse précédente. Sr ce stump, quelle est la classe attribuée à chaque feuille ? Quelle est son erreur sur l'échantillon d'apprentissage ?

2. Soit l'échantillon de test T donné dans le tableau ci-contre :

	a_1	a_2	a_3	classe
x_1	v_1	v'_1	v''_1	+
x_2	v_1	v'_2	v''_1	+
x_3	v_2	v'_1	v''_2	+
x_4	v_3	v'_2	v''_2	+
x_5	v_1	v'_2	v''_2	−
x_6	v_2	v_1	v''_1	−
x_7	v_2	v_1	v''_2	−
x_8	v_3	v'_1	v''_1	+

   (a) Indiquer la matrice de confusion du classifieur *stump* issu du test t_1 à la racine, sur les exemples de T.

   (b) Quelle est l'erreur du choix de t_1 telle que calculée sur T ?

   (c) Quel est son rappel des positifs ?

2 Régression (7 points + 1 pt bonus)

Un hypermarché dispose de 20 caisses. On s'intéresse au temps moyen d'attente en fonction du nombre de caisses ouvertes un jour de semaine. Le tableau ci-après donne x le nombre de caisses ouvertes et y le temps moyen d'attente correspondant.

x (nombre de caisses ouvertes)	3	5	8	10	12
y (temps moyen d'attente en minutes)	16	9.6	6	4.7	4

1. Calculer $\bar{x}$ la moyenne du nombre de caisses ouvertes et $\bar{y}$ la moyenne des temps d'attente moyen.
2. Placer les couples (x, y) sur un graphique adéquat. Observe-t-on un corrélation linéaire entre le nombre de caisses ouvertes et le temps d'attente moyen ?
3. Calculer f, la droite de régression $y = f(x)$ permettant d'expliquer y par x, avec l'estimateur des moindres carrés, sur la base de l'échantillon. Placer cette droite sur le graphique précédent.
4. Quelle est l'erreur quadratique moyenne de f sur les données de l'échantillon ?
5. **(Bonus 1 point)** À partir de ces données, estimer analytiquement le temps d'attente moyen en caisse lorsque 18 caisses sont ouvertes.

3 Questions de cours (1 point + 1 pt bonus)

1. Quels sont (1) la différence principale et (2) le principe commun, entre l'algorithme des k plus proches voisins et celui des k-moyennes ?
2. **Bonus 1 point)** Quelles sont les causes possibles du sur-apprentissage ?

4 Compréhension de code (5 points + 1 pt bonus)

Soit le programme python page suivante.

1. Expliquer ce que font les lignes de code terminée par # et un numéro (ne pas recopier le code sur votre copie, son numéro).
2. Expliquer en 3 phrases au maximum ce que fait globalement ce programme.
3. **(Bonus 1 point)** Dessiner le type de graphique obtenu à la fin .

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split          #1
from sklearn.tree import DecisionTreeClassifier               #2
from sklearn.model_selection import cross_val_score

df = pd.read_csv('foot.csv')
X = df.drop(['class'], axis=1)                                #3
X_train, X_test, y_train, y_test =
    train_test_split(X, y, test_size=0.30, random_state=42)   #4
clf = DecisionTreeClassifier(random_state=42)
max_depths = range(2,15)
reussite_md =[]
reussite_md_std = []

for i in max_depths:
    clf.set_params(max_depth=i)                               #5
    reussite_md_cv = cross_val_score(clf, X_train, y_train, cv=5)
    reussite_md.append(reussite_md_cv.mean())                 #6
    reussite_md_std.append(reussite_md_cv.std())

plt.figure()
plt.errorbar(np.array(max_depths), np.array(reussite_md),     #7
    yerr=np.array(reussite_md_std))
plt.title("Evolution du taux de réussite avec max_depth")
plt.xlabel("Valeur de max_depth")
plt.ylabel("Taux de réussite")
plt.show()
```