Matrices de Confusion

Nous allons durant cette séance utiliser les matrices de confusion et les mesures d'évaluation des résultats des tests de classifieurs de *Rappel*, *Précision*, *Spécificité* et *Taux de Faux Négatifs* afin de quantifier numériquement l'importance des différents types de succès et d'échecs de chacun, et ainsi optimiser le choix du classifieur en fonction des *objectifs de l'application*.

Rappel : afin de réaliser les exercices, référez-vous si besoin aux corrections des travaux dirigés précédents disponibles sous forme de scripts R sur la page du cours. Si la commande ou fonction à utiliser pour réaliser un exercice ne vous est pas indiquée dans l'énoncé, cela signifie qu'elle a déjà été utilisée.

1. Ensemble de données *Produit*

Caractéristiques de l'ensemble de données :

- Instances : 600 clients
- Nombre de variables : 12
- Valeurs manquantes : aucune
- Séparateur de colonnes : virgule
- Séparateur de décimales : point
- Variable de classe : Produit
- Variables prédictives : Age, Sexe, Habitat, Revenus, Marie, Enfants, Voiture, Compte_Epargne, Compte_Courant, Emprunt

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
Produit	Booléen	Client acquéreur du produit Variable de classe	Oui, Non

2. Chargement des données

- Chargez les données du fichier `Data Produit.csv` dans un data frame `produit`.
- Vérifiez le chargement des données dans le data frame `produit` en affichant la liste des variables et leur mode à l'aide de la fonction `str()`.
- Construisez l'ensemble d'apprentissage `produit_EA` et l'ensemble de test `produit_ET` comme suit :
  - `produit_EA` : sélection des 400 premières lignes de `produit`.
  - `produit_ET` : sélection des 200 dernières lignes de `produit`.

3. Installation des librairies

Nous allons utiliser les librairies R suivantes durant cette séance :

Librairie	Fonction
`rpart` (Recursive Partitioning and Regression Trees)	`rpart()`
`rpart.plot` (Plotting rpart trees)	`prp()`
`C50` (Decision Trees and Rule-Based Models)	`C5.0()`
`tree` (Classification and Regression Trees)	`tree()`

Rappel : afin d'installer et activer des librairies, vous pouvez utiliser :

- soit l'interface de RStudio (menu Tools pour les installer et onglet Packages pour les activer),
- soit les fonctions `install.packages("`*`nom_librairie`*`")` et `library(`*`nom_librairie`*`)` en ligne de commande.

☛ Installez les librairies *rpart*, *C50* et *tree* (si celles-ci sont déjà installées, cela les mettra-à-jour si c'est nécessaire) et activez-les dans R.

4. Apprentissage des arbres de décision

☛ Construisez les trois arbres de décision suivants :

☞Arbre `tree1` avec la fonction `rpart()`

☞Arbre `tree2` avec la fonction `C50()`

☞Arbre `tree3` avec la fonction `tree()`

selon les paramètres suivants :

☞Ensemble de données : `produit_EA`

☞Variable de classe (à prédire) : `Produit`

☞Variables prédictives : `Age`, `Sexe`, `Habitat`, `Revenus`, `Marie`, `Enfants`, `Voiture`, `Compte_Epargne`, `Compte_Courant`, `Emprunt`

☞Variables non utilisées : `ID`

Rappel : le paramètre « `Produit ~ .` » indiquant à la fonction d'utiliser toutes les variables autres que la variable `Produit` comme variables prédictives, il est nécessaire de supprimer la variable `ID` de `produit_EA` pour qu'elle ne soit pas prise en compte si vous utiliser ce paramètre.

☛ Affichez graphiquement l'arbre `tree1` à l'aide de la fonction `prp()` de la librairie `rpart.plot`.

☛ Affichez graphiquement l'arbre `tree2` à l'aide de la fonction `plot.C5.0()` de la librairie `C50`.

☛ Affichez graphiquement l'arbre `tree3` à l'aide des fonctions `plot.tree()` et `text.tree()` de la librairie `tree`.

5. Tests des classifieurs et calcul des matrices de confusion

☛ Pour chacun des arbres de décision `tree1`, `tree2` et `tree3` :

☞Appliquez l'arbre à l'ensemble de test `produit_ET` et stockez le résultat dans un objet `test_tree`*`N`*.

☞Affichez le nombre de prédictions de l'arbre dans chaque classe à l'aide de la fonction `table()`.

La matrice de confusion est une table de contingence dans laquelle sont affichées en lignes les classes réelles (classe des exemples de test dans l'ensemble de test) et en colonnes les prédictions de classes par le classifieur.

Cette table peut être construite par la fonction `table(`*`lignes, colonnes`*`)` dans laquelle :

- le vecteur *`ligne`* sera le vecteur du data frame `produit_ET` contenant les classes réelles (c'est-à-dire le vecteur `produit_ET$Produit`),
- le vecteur *`colonnes`* sera le vecteur contenant le résultat de l'application du classifieur au data frame `produit_ET` (c'est-à-dire le vecteur `test_tree`*`N`*),.

☛ Construisez la matrice de confusion pour le classifieur `tree1` en stockant la table obtenue dans un

objet `mc_tree1` par la commande :

```
> mc_tree1 <- table(produit_ET$Produit, test_tree1)
```

- ☛ Construisez les matrices de confusion pour les classifieurs `tree2` et `tree3` en stockant les tables obtenues dans des objets `mc_tree2` et `mc_tree3`.
- ☛ Affichez les trois matrices de confusion obtenues à l'aide de la fonction `print()` afin de déterminer s'il existe des différences dans les types de succès et d'erreurs obtenus pour les trois classifieurs.

6. Mesures d'évaluation

Pour cette application de prédiction d'appétence, la classe ***Produit=Oui*** constituera la classe positive (prioritaire) et la classe ***Produit=Non*** constituera la classe négative.

La colonne des « `Oui` » sera donc la colonne des positifs :

• Vrais Positifs (VP) : exemples de classe réelle « `Oui` » prédits « `Oui` ».

• Faux Positifs (FP) : exemples de classe réelle « `Non` » prédits « `Oui` ».

La colonne des « `Non` » sera donc la colonne des négatifs :

• Vrais Négatifs (VN) : exemples de classe réelle « `Non` » prédits « `Non` ».

• Faux Négatifs (FP) : exemples de classe réelle « `Oui` » prédits « `Non` ».

- ☛ Identifiez les nombres de Vrais Positifs, Faux Positifs, Vrais Négatifs et Faux Négatifs.
- ☛ En supposant que l'on souhaite minimiser le nombre de prédiction positives incorrectes, lequel de ces trois classifieurs obtient le plus faible nombre de Faux Positifs ?
- ☛ En supposant que l'on souhaite minimiser le nombre de prédiction négatives incorrectes, lequel de ces trois classifieurs obtient le plus faible nombre de Faux Négatifs ?

Les valeurs des cellules d'une table de contingence peuvent être accédées par leur indice à l'aide l'opérateur *`nom_table[num_ligne, num_colonne]`* de la même manière que celles d'un data frame.

Par exemple, accéder à la 1ère cellule de la 2ème ligne de la table `mc_tree1` se fait par la commande : `mc_tree1[2,1]`.

- ☛ En accédant aux valeurs des matrices de confusion `mc_tree1`, `mc_tree2` et `mc_tree3`, calculez les mesures de *Classification Accuary* (taux de succès) des trois classifieurs.

  Quel est le classifieur le plus performant parmi les trois arbres de décision si on considère comme critère d'évaluation le taux de succès global ?

Intéressons-nous maintenant à la classe positive spécifiquement, afin de déterminer quel modèle réalise les prédictions positives les plus fiables et quel modèle réussi à discriminer le plus d'exemples de test positifs.

- ☛ Calculez pour chacune des trois matrices de confusion la mesure du *Rappel* (*Sensibilité*) – dont la formule de calcul est : VP / (VP+FN) – permettant d'évaluer la capacité du classifieur à détecter les exemples positifs de l'ensemble de test.

  Quel est le classifieur le plus performant parmi les trois arbres de décision selon le critère d'évaluation du *Rappel*?
- ☛ Calculez pour chacune des trois matrices de confusion la mesure de *Précision* – dont la formule de calcul est : VP / (VP+FP) – permettant d'évaluer la fiabilité des prédictions positives du classifieur.

  Quel est le classifieur le plus performant parmi les trois arbres de décision selon le critère d'évaluation de la *Précision*?

Considérons maintenant la classe négative, afin de déterminer quel modèle réalise les prédictions négatives les plus fiables et quel modèle réussi à discriminer le plus d'exemples de test négatifs.

- ☛ Calculez pour chacune des trois matrices de confusion la mesure de *Spécificité* – dont la formule de calcul est : VN / (VN+FP) – permettant d'évaluer la capacité du classifieur à détecter les exemples négatifs de l'ensemble de test.

  Quel est le classifieur le plus performant parmi les trois arbres de décision selon le critère d'évaluation de la *Spécificité*?
- ☛ Calculez pour chacune des trois matrices de confusion la mesure du *Taux de Vrais Négatifs* – dont la formule de calcul est : VN / (VN+FN) – permettant d'évaluer la fiabilité des prédictions négatives du classifieur.

  Quel est le classifieur le plus performant parmi les trois arbres de décision selon le critère d'évaluation

du *Taux de Vrais Négatifs*?