Courbes ROC et Indices AUC

Nous allons durant cette séance comparer différents classifieurs à l'aide des représentations graphiques de leurs performances par les courbes ROC et leur indicateur AUC associé. L'objectif est d'identifier le classifieur le plus performant parmi plusieurs modèles de prédiction d'appétence des clients.

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 durant cette séance utiliser les librairies R suivantes :

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()`

`ROCR` (Visualizing the Performance of Scoring Classifiers)	`prediction()`
	`performance()`
	`plot.performance()`

- Installez ces librairies et activez-les dans R.
- Affichez les documentations de ces librairies soit via l'onglet *Packages* dans la zone bas-droite de RStudio, soit par la fonction `help()`, afin d'avoir un aperçu des fonctions fournies par chacune de ces librairies.

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 cible : `Produit`
  - Variables prédictives : `Age`, `Sexe`, `Habitat`, `Revenus`, `Marie`, `Enfants`, `Voiture`, `Compte_Epargne`, `Compte_Courant`, `Emprunt`
  - Variables non utilisées : `ID`
- 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. Calcul des courbes ROC

Afin de comparer les performances des trois arbres de décision sur l'ensemble de test, nous allons tracer la courbe ROC de chacun d'eux sur le même graphique.

Plusieurs librairies R fournissent des fonctions de calcul de courbes ROC et des indicateurs de performance AUC (Area Under Roc Curve) ou Coefficient Gini.

Parmi celles-ci, nous allons utiliser la librairie *ROCR* qui fournit les fonctions `prediction()`, `performance()` et `plot.performance()` pour ces calculs.

- Consultez la documentation de la librairie *ROCR* afin d'avoir un aperçu des fonctions disponibles.
- Le traçage de la courbe ROC, et le calcul de l'indice AUC, nécessitent de disposer pour chaque exemple de l'ensemble de test de :
  - La classe réelle de l'exemple (variable `produit_ET$Produit`).
  - La probabilité de prédiction de l'exemple dans la classe positive (probabilité de prédiction « `Oui` »).

Rappel : les paramètres d'application de la fonction `predict()` dépendent du type de classifieur appliqué, c-à-d de la fonction de construction du classifieur qui a été utilisée.

Afin de générer les probabilités pour chaque classe (et non la classe prédite), la commande à utiliser pour les arbres créés par les fonctions `rpart()` et `C5.0()` est la suivante :

```
> prob_arbre <- predict(nom_arbre, data_frame, type = "prob")
```

Afin de générer les probabilités de prédiction dans chaque classe pour un arbre créé par la fonction

`tree()`, la commande à utiliser est :

```
> prob_arbre <- predict(nom_arbre, data_frame, type = "vector")
```

À partir des probabilités des prédictions faites sur l'ensemble de test `produit_ET` et de la classe réelle des exemples dans l'ensemble de test (variable `produit_ET$Produit`), nous allons générer les données nécessaires pour le calcul des taux de vrais positifs (TVP) et taux de faux positifs (TFP) qui serviront ensuite à tracer la courbe ROC.

- Générez pour chaque instance de l'ensemble de test les probabilités de prédiction des deux classes par l'arbre `tree1` avec la fonction `predict()` :

```
> prob_tree1 <- predict(tree1, produit_ET, type="prob")
```

- Générez les données nécessaires au calcul de la courbe ROC avec la fonction `prediction()` :

```
> roc_pred1 <- prediction(prob_tree1[,2], produit_ET$Produit)
```

- Calculez les taux de vrais positifs (`tpr`) et taux de faux positifs (`fpr`) avec la fonction `performance()` :

```
> roc_perf1 <- performance(roc_pred1,"tpr","fpr")
```

- Tracez la courbe ROC à partir des taux de vrais et faux positifs générés dans `roc_perf1`, en couleur verte, avec la fonction `plot.performance()` :

```
> plot(roc_perf1, col = "green")
```

- Tracez sur le même graphique les courbes ROC de chacun des deux autres arbres de décision `tree2` et `tree3` :
  - Suivez le même procédé de calcul des objets `prob_tree`*N*, `roc.pred`*N* et `roc_perf`*N* que pour `tree1`.
  - Afin d'ajouter la courbe ROC sur le même graphique au lieu d'en créer un nouveau, il est nécessaire d'ajouter le paramètre `add = TRUE` à la commande `plot()`. Par exemple, pour ajouter la courbe ROC du classifieur `tree2` (en bleu) au graphique contenant la courbe ROC de `tree1` il faut utiliser la commande : `plot(roc_perf2, col = "blue", add = TRUE)`
- À partir des courbes obtenues, identifiez le ou les arbres de décision les plus performants, c'est-à-dire ceux qui constituent l'enveloppe convexe du graphique.

6. Calcul des indices AUC

Nous allons maintenant calculer l'indice AUC (Area Under the ROC Curve) de chacun des arbres de décision afin d'obtenir une mesure numérique d'évaluation des performances globales de la distinction de la classe positive pour chacun de ces trois arbres.

La mesure d'AUC calcule quel est la proportion représentée par la surface entre la courbe ROC et l'axe des abscisses par rapport à la surface totale du graphique.

Nous allons utiliser la fonction `performance()` de la librairie *ROCR* pour ce calcul.

- Calculez l'indice de performance AUC pour l'arbre `tree1` à partir des données générées dans `roc_pred1` avec la fonction `performance()` par la commande :

```
> auc_tree1 <- performance(roc_pred1, "auc")
```

- L'objet `auc_tree1` généré est un objet spécifique constitué de plusieurs attributs dont la liste peut être affichée par la commande :

```
> str(auc_tree1)
```

- Afficher la valeur de la mesure d'AUC qui est stockée dans l'attribut `y.values` de l'objet `auc_tree1` à l'aide de la fonction `attr()` par la commande :

```
> attr(auc_tree1, "y.values")
```

- Calculez l'indice de performance AUC pour chacun des autres arbres de décision.
- À partir des valeurs obtenues, identifiez l'arbre de décision ayant l'indice AUC le plus important.