Arbre de décision pour la prédiction de maladies cardiovasculaires

Projet d'introduction aux datasciences

Projet à rendre avant 03/09/2019

M1 santé publique *Université Paris-Sud*

---

1. Jeu de données

Dans ce projet, nous allons faire appel au jeu de données de maladies cardiovasculaires disponible sur le site. Il est fortement recommandé de visiter le site et de parcourir le descriptif du jeu de données. Comme vous pouvez le constater, le dossier de données contient différents fichiers. Nous utilisons pour ce projet le fichier processed.cleveland.data. Commençons par le lecture du jeu de données comme suit

```
cardio <- read.csv("processed.cleveland.data", header = FALSE, na.strings = '?')
names(cardio) <- c( "age", "sex", "cp", "trestbps", "chol",
                    "fbs", "restecg", "thalach","exang",
                    "oldpeak","slope", "ca", "thal", "status")
```

Nous souhaitons ajuster un modèle d'arbre de décision pour prédire les valeurs sous le statut de variable réponse présente à la 14ième colonne du jeu de données, le **statut de maladie angiographique** qui identifie ou classe chaque patient comme "ayant une maladie cardiaque" ou "n'ayant pas de maladie cardiaque. Intuitivement, nous nous attendons à ce que certaines (ou toutes) les autres 13 variables nous aident à prédire la variable réponse.

Commençons par nous familiariser avec les données. Pour cela:

1. Appliquer la fonction str au data frame cardio. Les formats des variables du jeu de données sont-ils satisfaisants ?
2. À l'aide de la fonction factor, transformer les variables qualitatives du jeu de données. Vérifier que les variables sont aux bons formats.
3. Quelles sont les modalités de la variable réponse ?
4. Transformer la variable réponse status en une variable binaire où status="0" si le patient n'est pas atteint d'une maladie cardiaque et status="1" sinon. Appliquer la fonction summary au jeu de données.
5. Afficher le nombre de données manquantes du jeu de données. Écarter les données manquantes à l'aide de la fonction na.omit.

Maintenant, on peut passer à la modélisation.

6. Fixer la graine du générateur aléatoire à 1 à l'aide de la fonction set.seed et répartir le jeu de données en jeux de données cardio.train et cardio.test de tailles respectives de 70% et 30%.
7. Ajuster, tracer et comparer en terme d'erreur de test les arbres de décisions obtenus respectivement sans et avec élagage. Afficher l'importance relative de chaque variable explicative du modèle dans la prédiction de la variable réponse.