

DÉTECTION DE MAUVAISE SPÉCIFICATION D'UN MODÈLE PARAMÉTRIQUE

Vous aurez besoin pour ce TP des librairies R suivantes :

rpart, caret, pROC, ISLR

Partie 1 : Un premier problème de classification.

L'objectif est ici de comparer un modèle paramétrique simple (en l'occurrence la régression logistique), donné par

$$\mathbb{P}(Y = 1 | X = x) = G(\beta_0 + x^T \beta),$$

avec G la fonction de répartition d'une loi logistique ; à un modèle non-paramétrique beaucoup plus général donné par

$$\mathbb{P}(Y = 1 | X = x) = G(h(x)),$$

où les fonctions G et h sont inconnues (les modèles de *bagging*, *forêts aléatoires* et *boosting* permettent d'estimer ce modèle général, par des algorithmes différents).

Notez que les sources d'erreur de la première modélisation peuvent être de deux ordres :
i) la spécification de la forme linéaire, ii) le choix de la loi G .

Nous ferons ici une analyse et une comparaison des résultats sur la base d'une validation croisée à 10 blocs.

- (1) Importer la librairie **ISLR** et charger le jeu de données *Carseats* dans R.
- (2) Comprendre cette base de données. Construire une variable binaire permettant de rendre compte d'un effectif de vente (**Sales**) supérieure à 8 pour chaque ligne de la base. C'est cette variable qui sera à expliquer dans notre problème de classification.
- (3) Afficher un résumé des données. Quelle est la proportion de bonnes ventes (>8) ?
- (4) Modélisation 1 : on va utiliser la fonction **train** et la méthode **glm** pour créer un modèle de régression logistique intégré à une procédure de validation croisée.
 - i) Comprendre l'utilisation de *rpart* : consulter l'aide de la fonction et...
 - a) : ...comprendre ses arguments et ce qu'elle retourne (attributs de l'objet) ;
 - c) : ...comprendre sur quels paramètres vous jouerez en tant qu'utilisateur ;
 - ii) Implémenter l'algorithme : expliquer l'espèce d'iris à partir des mesures sur les pétales et sépales.
 - a) construire un modèle **optimal**, et afficher les résultats dans le terminal R en tapant le nom de l'objet créé ;
 - b) interpréter les résultats affichés : retrouver les taux d'erreur et les prévisions par noeud. Combien cet arbre a-t-il de feuilles ?
 - d) accéder aux attributs de l'objet créé. Combien y en a-t-il ? Les comprendre un par un. Finalement, quelle variable explicative est la plus importante ?
 - i) effectuer des prévisions pour un jeu quelconque de caractéristiques sur les pétales et sépales.
- (5) Modélisation 2 : ...

Partie 2 : Application à une problématique de régression.

- (1) Charger le jeu de données *cu.summary* dans le terminal R.
- (2) Consulter l'aide des données pour les comprendre, et visualiser. Quelle dimension ?
- (3) Afficher un résumé des données. Nous tenterons d'expliquer le prix des véhicules en fonction de l'ensemble des autres caractéristiques fournies.
- (4) Construire l'arbre maximal : quel argument de la fonction `rpart` faut-il absolument modifier par rapport à précédemment ?
- (5) Afficher les résultats numériques et interpréter : retrouver par exemple la valeur de l'homogénéité de la racine. À quoi correspond cette valeur ? Retrouver également la prévision du prix dans la racine.
- (6) Afficher graphiquement l'arbre maximal.
- (7) Poursuivre les mêmes étapes que dans la première partie pour son élagage, en illustrant le phénomène de surapprentissage.
- (8) De quelle taille est votre arbre optimal ? Quelle est (en pourcentage) la variance non-expliquée par votre estimateur ? Interpréter opérationnellement vos résultats.