

TP : Supervisée

Vous pouvez faire les exercices dans le langage de votre choix.

1 Exercice 1 : Position des joueurs NFL

Dans cet exercice, on se propose de développer un modèle permettant de classifier les joueurs NFL grâce à l'analyse discriminante.

1. Télécharger le jeu de données suivant : [lien](#).
2. Faire une rapide analyse descriptive des données.
3. Faire une analyse discriminante pour classer les joueurs des positions QB (quart-arrière) et OL (ligne offensive) en utilisant leur taille (Ht) et leur poids (Wt).
4. Quelle serait la coordonnée d'un joueur ayant une taille est de 76.5 et un poids de 335.5 sur le nouvel axe ? Quelle serait la position d'un tel joueur ?
5. Faire la même chose avec les joueurs des lignes offensives (OL) et défensives (DL), puis entre les joueurs de la ligne défensive (DL) et les quarts-arrières (QB). En utilisant les résultats de ce modèle, quelles positions sont les plus faciles à séparer avec l'analyse discriminante ?
6. Faire une analyse discriminante pour classer les joueurs des positions QB, OL et DL. La classification des joueurs est-elle facilitée avec ce modèle ou bien est-il préférable d'utiliser les trois modèles précédents ?

2 Exercice 2 : Prédire les réclamations

Dans cet exercice, on se propose de construire un arbre de classification permettant de prédire si une réclamation pourrait avoir lieu sur un colis envoyé par Rakuten. Le jeu de données fourni par PriceMinister Rakuten contient une variable cible (CLAIM_TYPE), ainsi que 12 variables explicatives.

1. Télécharger le jeu de données suivant : [lien](#).

2. Faire une rapide analyse descriptive des données.
3. Partitionner les données en échantillon d'entraînement (70%), de validation (30%) de façon aléatoire.
4. Ajuster un arbre de classification sur le jeu d'entraînement en utilisant les paramètres par défaut.
5. Faire une prédiction sur les jeux d'entraînement et de validation.
6. Estimer le taux d'erreur dans le noeud 3.
7. Calculer le taux d'observations bien classifiés global sur les échantillons d'entraînement et de validation.
8. Changer les hyperparamètres tel que l'effectif minimal pour q'un noeud puisse être séparé soit de 2 et l'effectif minimal d'un noeud terminal soit de 1 et reconstruire l'arbre.
9. Faire varier le paramètre de complexité du modèle et analyser les arbres résultant.
10. Calculer le taux d'erreur des différents modèles sur les échantillons d'entraînement et de validation. Tracer le graphique du taux d'erreur en fonction de la complexité du modèle.

3 Exercice 3 : Un *bagging* de l'analyse discriminante et de l'arbre de classification

Dans cet exercice, on se propose de simuler des données pour regarder les performances du *bagging* des algorithmes d'analyse discriminante et d'arbres de classification. Considérons une première classe ayant pour distribution $Y_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$ et une deuxième classe ayant pour distribution $Y_2 \sim \mathcal{N}(\mu_2, \Sigma_2)$ avec

$$\mu_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 1 & 1 \\ 1 & 4 \end{pmatrix}, \quad \text{et} \quad \mu_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 4 & 4 \\ 4 & 16 \end{pmatrix}.$$

1. Générer un jeu de données d'entraînement ($n = 200$) et un jeu de données de test ($n = 1000$) ayant la distribution décrite précédemment.
2. Implémenter le *bagging* de l'analyse discriminante. Vérifier les performances de modèle en l'entraînant sur le jeu d'entraînement et en calculant son erreur sur le jeu de test pour un nombre différent de classificateurs de base, par exemple sur une grille de vingt valeurs comprises entre 0 et 500. Tracez le graphique de l'erreur en fonction du nombre de classificateurs de base. Interpréter le graphique.
3. Faire la même chose en utilisant un arbre de classification comme classificateurs de base. On pourra aussi regarder l'influence de la taille minimale d'une feuille pour faire la découpe. Interpréter les résultats.

4 Exercice 4 : Est-ce que j'ai reçu un spam ?

Dans cet exercice, on cherche à prédire si un mail reçu est un spam ou non à l'aide d'une forêt aléatoire.

1. Télécharger le jeu de données suivant : [lien](#). Une description du jeu de données est [ici](#).
2. Partitionner les données en échantillon d'entraînement (50%), de validation (50%) de façon aléatoire.
3. Construire une forêt aléatoire en laissant les paramètres par défaut. Interpréter les résultats.
4. Explorer l'influence du nombre de variables choisies aléatoirement à chaque noeud sur les performances du modèle. Interpréter les résultats.
5. Utiliser la validation croisée pour choisir le nombre optimal de variables choisies aléatoirement à chaque noeud.
6. Explorer l'influence du nombre d'arbres dans la forêt. Tracer l'erreur de classification en fonction du nombre d'arbres (prendre une grille de 1 à 50 pour le nombre d'arbres) en choisissant 1, 7 et 35 variables aléatoirement à chaque noeud. Interpréter les résultats.
7. Pour une forêt aléatoire en choisissant une variable aléatoirement à chaque noeud, tracer l'erreur de classification pour une forêt aléatoire ayant 1, 2, ..., 50 arbres mesuré sur le jeu de test et sur les données non-utilisées dans les échantillons *bootstrap* pour la création du modèle (ce que l'on appelle *out-of-bag error*).

5 Exercice 5 : Comparaison des performances de classificateurs.

Dans cet exercice, on cherche à comparer les performances des différents classificateurs que l'on a vu pendant le cours.

1. Télécharger le jeu de données suivant : [lien](#). Une description du jeu de données est [ici](#).
2. Évaluer les performances de l'analyse discriminante, d'un arbre de classification, d'une forêt aléatoire, de *adaboost* et de *xgboost* en découpant le jeu de données en ensemble d'entraînement (70%) et de test (30%).
3. Répéter ce processus 100 fois. Tracer les boxplots des erreurs de classification pour chacune des méthodes.
4. Interpréter les résultats.