Nous travaillons sous R. Tous les tests sont à 5%.

1 Supports

Le site de notre cours est http://eric.univ-lyon2.fr/~ricco/cours/cours_regression_logistique.html

Plus spécifiquement pour cette séance, nous nous référerons à :

TUTO 1 – http://tutoriels-data-mining.blogspot.com/2012/03/introduction-r-regression-logistique.html

TUTO 2 – http://eric.univ-lyon2.fr/~ricco/cours/cours/pratique_regression_logistique.pdf

TUTO 3 – http://eric.univ-lyon2.fr/~ricco/cours/slides/filtrage_predicteurs.pdf

TUTO 4 - http://tutoriels-data-mining.blogspot.com/2018/04/machine-learning-avec-caret.html

N'oubliez pas que vous avez accès aux corrigés de nos précédentes séances !

2 Données

Nous travaillons sur la base « Chess (King-Rook vs. King-Pawn) ». Elle décrit les configurations gagnantes ou perdantes sur un échiquier selon les positions des pièces.

Le fichier « **kr-vs-kp.txt** » est au format CSV (séparateur tabulation). On cherche à expliquer la « classe » $\in$ {« nowin », « won »} à partir des autres descripteurs disponibles. Toutes les variables sont qualitatives.

3 Exercices – Sélection de variables

3.1 Fonction d'évaluation des prédictions

0. Ecrivez une fonction qui prend en entrée deux vecteurs, la classe observée (y_obs) et la classe prédite (y_pred) fournie par un modèle quelconque. Voici le prototype de la fonction : `taux_erreur <- function(y_obs,y_pred){…}`. Elle doit calculer et afficher :
    a. La matrice de confusion (en ligne classe observée) (table).
    b. Le nombre d'observations mal classées.
    c. Le taux d'erreur.
    d. L'intervalle de confiance à 90% du taux d'erreur (cf. par ex. pour le calcul de l'intervalle de confiance d'une proportion : https://webapps.fundp.ac.be/umdb/biostats2017/biostat/modules/module105/page4.html).

3.2 Importation et partition des données

1. Importez le fichier « **kr-vs-kp.txt** » (read.table). Attention, le caractère tabulation fait office de séparateur de colonne (sep), la première ligne contient les noms des variables (header).

2. Affichez les caractéristiques du data frame (str). Quel est le nombre d'observations et de variables ? (3196 obs., 35 variables).

3. Nous souhaitons partitionner ces données en échantillons d'apprentissage (**2196 obs.**) et de test (**1000 obs.**). Pour ce faire, nous nous inspirons du tutoriel (les lignes 8 à 12) : https://gist.github.com/duttashi/9208fb66142406d0ece74ba31cf181aa

    a. Initialisez le générateur de nombre aléatoire pour que nous ayons tous les mêmes partitions set.seed(1)
    b. Créez un index qui représente les indices des observations à intégrer dans l'échantillon d'apprentissage (sample, attention size = 2196 pour nous).
    c. Construisez le data frame des individus en apprentissage avec cet index.
    d. Construisez le data frame des individus en test avec la négation de cet index.
    e. Contrôlez les dimensions des deux data frame générés [(2196, 35) en apprentissage, (1000, 35) en test].

3.3 Modélisation avec la totalité des variables

4. Nous souhaitons prédire la variable « classe ». Nous construisons le modèle sur les données d'apprentissage en intégrant toutes les autres variables disponibles (glm). Juste pour contrôler nos résultats, quelle est la valeur de l'AIC ? (460). Que constatez-vous concernant les variables utilisées dans la régression ? (elles ont été binarisées en indicatrices 0/1, l'identifiant de la seconde modalité a été accolé au nom de la variable)

5. Effectuez la prédiction sur l'échantillon test (**TUTO 1**, page 10) et calculez les indicateurs de performances par l'entremise de la fonction taux_erreur() rédigée ci-dessus. Quel est le nombre d'observations mal classées (33), le taux d'erreur estimé (0.033), son intervalle de confiance à 90% ([0.0237 ; 0.0423]).

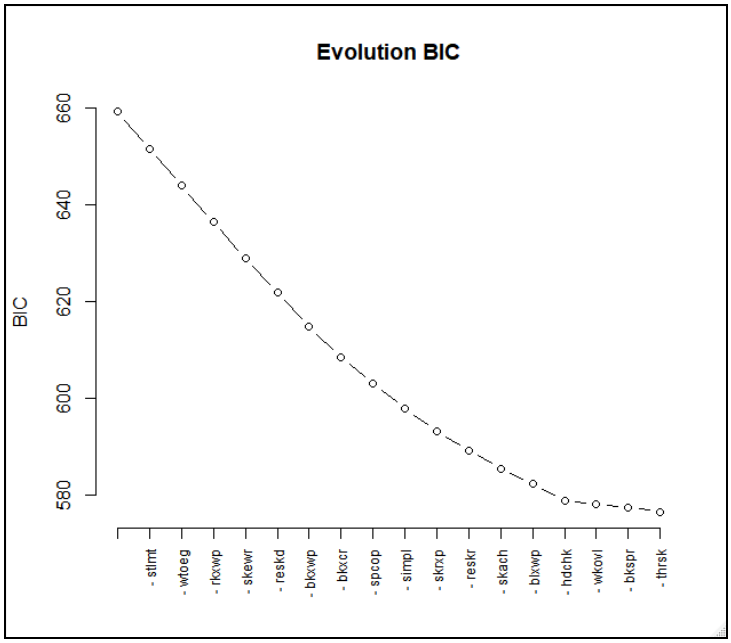
3.4 Sélection de variables stepAIC

Nous souhaitons utiliser une procédure de sélection de variables en lien direct avec l'algorithme d'apprentissage dans un premier temps, on parle de processus de sélection « intégrée » (embedded). Nous faisons appel à la fonction stepAIC (package « MASS ») qui permet, via un algorithme glouton, de réaliser une sélection ascendante (forward) ou descendante (backward) en optimisant le critère AIC (ou BIC selon le paramètre sélectionné).

6. Chargez la librairie « MASS » (library).

7. Faites appel à stepAIC pour réaliser une **sélection backward** (**TUTO 1**, page 11), en optimisant le critère BIC (voir la doc de stepAIC pour fixer correctement la valeur du paramètre k de stepAIC permettant de manipuler le critère BIC).

8. Combien de variables ont été sélectionnées par le processus ? (17) (cf. la propriété $coefficients peut-être).
9. Donnez la liste de ces variables (bkblk, bknwy, bkon8, …., wkpos) (il faut prendre les noms des variables brutes, sans les modalités que R leur associe automatiquement lors du recodage dans la régression).
10. Quelles sont les variables retirées ? Dans quel ordre l'ont-elles été ? (stml, wtoeg, rkxwp, …, thrsk).
11. Affichez un graphique montrant l'évolution du critère BIC au fur et à mesure du retrait des variables. Il devrait ressembler à ceci (cf. https://stackoverflow.com/questions/5182238/replace-x-axis-with-own-values pour la manipulation des axes d'un graphique)



12. Quelles sont les performances en test du modèle issu de la sélection backward ? (intervalle de confiance de l'erreur [0.0289 ; 0.049]).
13. Est-ce que ce modèle se démarque significativement du précédent ?
14. Réitérez la sélection en vous appuyant sur un algorithme forward cette fois-ci (**TUTO 1**, page 12), toujours en optimisant le critère BIC. Combien de variables sont sélectionnées ? (17 aussi) Lesquelles ? (rimmx, bxqsq, wknck, …, thrsk) Quel est l'ordre d'introduction des variables ? (rimmx, bxqsq, wknck, …) Quelles sont les performances en test ? (taux d'erreur ϵ [0.0239 ; 0.049])

15. Les deux approches fournissent le même nombre de variables et présentent les mêmes performances en test. Mais est-ce que les sous-ensembles de variables sélectionnées sont identiques ?

    a. Combien et quelles variables ont-elles en commun ? (15) (intersect)
    b. Quelles sont les variables présentes dans forward et qui ne le sont pas dans backward ? (2 ; bkspr, thrsk) (setdiff)
    c. Inversement ? (2 ; dsopp, wkcti)
    d. Que faut-il penser de tout ceci ?

3.5 Méthodes filtres pour la sélection de variables

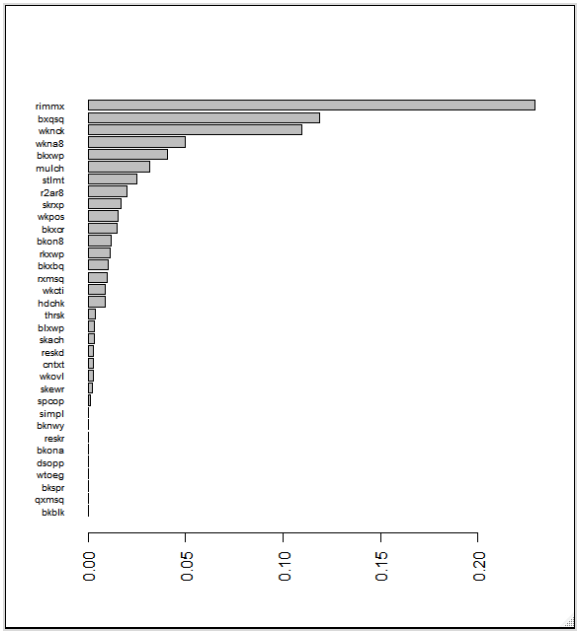
Les méthodes filtres agissent en amont des méthodes c.-à-d. elles sélectionnent les variables selon des critères intrinsèques, en faisant l'hypothèse que le sous-ensemble sélectionné convient pour tout algorithme de machine learning mise en œuvre par la suite (**TUTO 3**). Elles ont pour avantage la rapidité, mais pour inconvénient une « universalité » qui n'est pas toujours de bon aloi.

16. Installez et chargez la librairie « FSelector ».

17. On souhaite impmlémenter l'apprche « ranking » (**TUTO 3**, page 8 et suivantes). Calculez l'importance des variables selon le critère « Symmetrical Uncertainty » (SU) (symmetrical.uncertainty).

18. Présentez les résultats triés de manière décroissante selon le critère SU (rimmx, bxqsq, wknck, wkna8, …)

19. Affichez un barplot() énumérant les variables par ordre décroissant d'importance.



20. Récupérez la liste des 17 meilleures variables au sens du critère SU (cutoff.k). Quelles variables a-t-elle en commun avec celle issue de la sélection backward ? (11 variables) Avec la sélection forward ? (10 variables)
21. Définissez la formule « classe ~ var1 + var2 + etc. » à partir de la liste sélectionnée par SU (as.simple.formula)
22. Créez le modèle de régression logistique (glm) avec cette formule. Quelles sont les performances en test ? ([0.036 ; 0.058]) Que faut-il en penser ?
23. Nous souhaitons passer à la sélection CFS qui gère les redondances entre les prédictives en optimisant le critère MERIT (**TUTO 3**, page 14). Quelles sont les variables sélectionnées avec cette approche (cfs) ? (3 ; bxsqs, rimmx, wknck)
24. Réalisez la régression sur ces variables et évaluez le modèle sur l'échantillon test. Que constatez-vous ? (103 individus mal classés, [0.087 ; 0.118])

3.6 Ranking + Wrapper

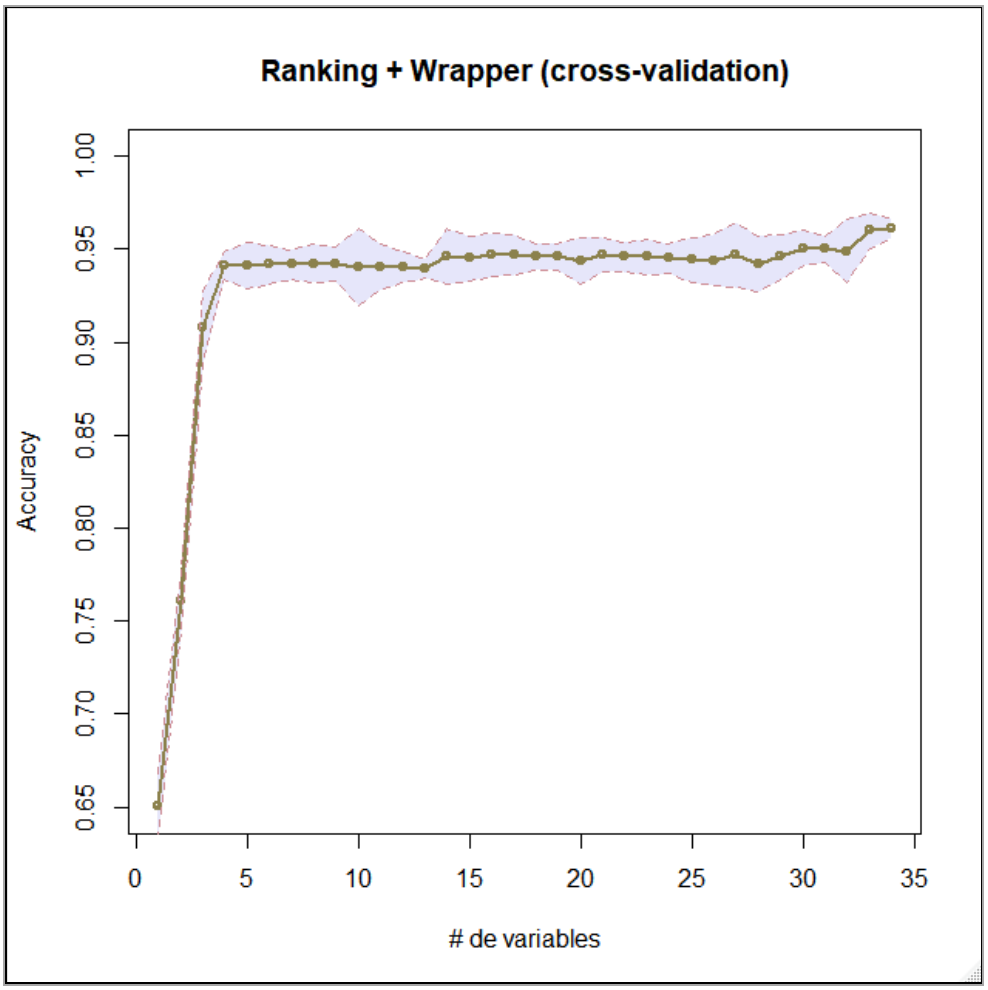
« Wrapper » est une méthode de sélection qui s'appuie explicitement sur les performances prédictives des modèles pour sélectionner le sous-ensemble « optimal » de variables. Il est en lien direct avec les caractéristiques de l'algorithme d'apprentissage, mais (1) il est souvent très gourmand en temps de calcul et (2) propice au surapprentissage, nécessitant un lancement répété de l'algorithme de machine learning (http://tutoriels-data-mining.blogspot.com/2009/05/strategie-wrapper-pour-la-selection-de.html).

Pour atténuer ces deux inconvénients, on se propose de tirer parti de l'ordonnancement des variables établi par le ranking selon SU pour évaluer les performances des scénarios imbriqués de combinaison de variables (1 variable, 2 variables, etc., toutes les variables).

25. Installez et chargez la librairie « caret ».
26. Préparez une structure matricielle avec 2 colonnes et autant de lignes qu'il y a de variables prédictives candidates dans la base.
27. En bouclant sur les combinaisons imbriquées des variables définies par le ranking selon SU :
    a. Lancez les régressions logistiques en demandant une validation croisée à 5 folds (voir http://eric.univ-lyon2.fr/~ricco/cours/slides/resampling_evaluation.pdf, page 11 pour le principe de la validation croisée ; **TUTO 4**, au bas de la page 5 – fonction train() – pour le lancement de la régression sauf qu'il faut spécifier method = « cv » et number = 5 dans le trainControl).
    b. Collectez le taux de reconnaissance (Accuracy) et son écart-type (AccuracySD) dans la structure matricielle préalablement préparée (voir la documentation de train() du

package « caret » pour comprendre comment récupérer les résultats dans le champ $results).

28. Construire alors un graphique permettant de suivre l'évolution de l'accuracy à ± 1 écart-type pour identifier la combinaison la plus performante et son gap par rapport aux autres solutions potentielles. Vous devriez obtenir un graphique ressemblant à ceci (voir https://stackoverflow.com/questions/14069629/how-can-i-plot-data-with-confidence-intervals pour dessiner un intervalle de confiance dans un graphique)



29. Quelle est le nombre de variables le plus performant au sens de cette démarche ?