Master 2 MMS
Apprentissage automatique

TP6 : Forêts aléatoires

Exercice 1. Récupérer le jeu de données d'apprentissage habituel synth_train.txt. On a $Y \in \{1, 2\}$ et $X \in \mathbb{R}^2$. On dispose de 100 données d'apprentissage.

1. Charger le jeu de données dans R. Transformer la variable de sortie `y` en facteur.
2. Charger le package `randomForest`. Construire une forêts aléatoire à l'aide de la fonction `randomForest` en gardant les paramètres par défaut. Consulter l'aide de la fonction `randomForest` notez bien tous les paramètres par défaut afin de savoir exactement quel algorithme est appliqué.
3. Afficher les résultats et vérifiez que vous comprenez les sorties associée au `print` de la forêt. Vérifiez ensuite que vous comprenez les éléments suivants de la forêt : `predicted`, `confusion`, `importance`, `importanceSD`. Testez la fonction `varImpPlot`. Regardez ensuite les éléments `votes`, `oob.times` et leur lien avec l'argument `norm.votes`.
4. Vérifiez ensuite que vous comprenez l'élément `err.rate` de la forêt. Retrouver dans `err.rate` le taux d'erreur OOB global et par classe obtenus avec la fonction `print`. Utiliser ensuite les résultats présents dans `err.rate` pour faire un graphique donnant une idée du calibrage du paramètre `ntree`.
5. Calculer le taux d'erreur d'apprentissage.
6. Charger le jeu de données test synth_test.txt puis calculer le taux d'erreur test de la forêt paramétrée par défaut.
7. Modifiez le paramètrage de la fôret pour que la méthode d'ensemble utilisée soit le `bagging`. Calculer alors le taux d'erreur des données test.

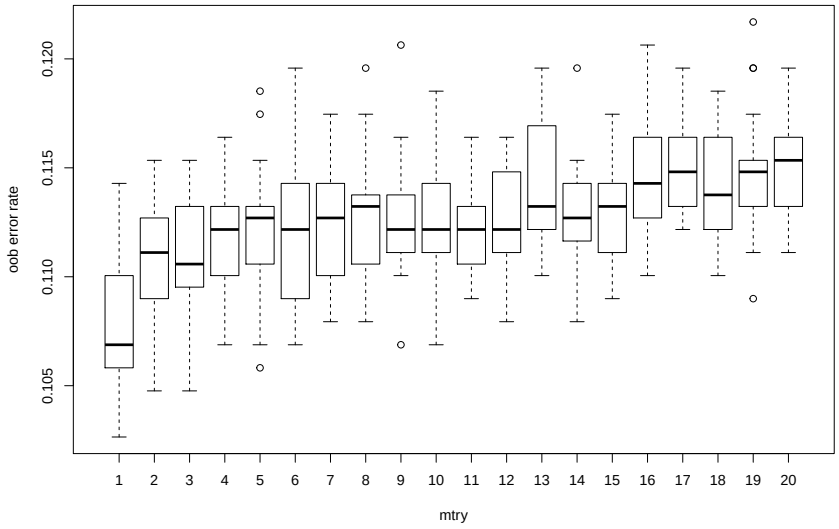
Exercice 2. On reprend les données concernant $n = 1260$ exploitations agricoles. Les variables explicatives sont $p = 30$ critères économiques et financiers et la variable qualitative à expliquer est la variable difficulté de paiement (0=saine t 1=défaillant).

1. Charger le jeu de données Desbois_complet.rda dans R.
2. Créez un découpage aléatoire des données en 945 observations d'apprentissage et 315 observations test.

```
set.seed(10)
tr <- sample(1:nrow(data),945)
train <- data[tr,]
test <- data[-tr,]
```

3. Quelle est l'erreur OOB, l'erreur test de la forêt constuite sur les données d'apprentissage avec les paramètres `mtry` et `ntree` par défaut ?
4. Le nombre d'arbre par défaut vous semble-il suffisant ?
5. Afin d'avoir une première idée du choix du paramètre `mtry` reproduire le graphique ci-dessous.

```
boxplot(err_oob,ylab="oob error rate",xlab="mtry")
```



6. Proposez une procédure de choix automatique du paramètre `mtry` (utilisant l'erreur OOB).
7. Prédire les données test avec la valeur optimale de `mtry` obtenue à la question précédente et calculer le taux d'erreur test.
8. Comparez ensuite les performances des forêts aléatoires avec celles de la régression logistique pour ces données. Attention de bien inclure le calibrage automatique du paramètre `mtry` dans la procédure de comparaison.

Exercice 3. Refaire le traitement proposé dans cet article de blog concernant les *imbalanced data* (partie avec le package `caret`) :

`https://shiring.github.io/machine_learning/2017/04/02/unbalanced`