Université Paris Cité
UFR de Mathématiques et Informatique

Classification Master 1 IMB
Examen final du 5 mai 2022 - durée 2h

On écrira tous les codes R dans un document `Nom_Prenom.R` ou `Nom_Prenom.Rmd`. Les packages suivants seront utilisés : `ade4`, `DALEX`. Utiliser la commande `library` pour charger ces packages. Si un ou plusieurs de ces packages ne sont pas disponibles, utiliser la commande `install.packages` pour les installer.

Exercice 1

Soit le tableau de données suivant (5 observations et 4 variables binaires)

	X^1	X^2	X^3	X^4
e_1	1	1	0	1
e_2	0	0	1	0
e_3	1	0	1	1
e_4	0	0	1	1
e_5	1	0	0	0

1. Entrer le tableau de données en R et appliquer la fonction dist.binary de la librairie `ade4` pour calculer les dissimilarités de Jaccard entre les individus.

Correction.

```
X = data.frame(X1=c(1,0,1,0,1),X2=c(1,0,0,0,0),X3=c(0,1,1,1,0),X4=c(1,0,1,1,0))
library(ade4)
D = dist.binary(X,method=1)
D
```

renvoie

```
          1         2         3         4
2 1.0000000
3 0.7071068 0.8164966
4 0.8660254 0.7071068 0.5773503
5 0.8164966 1.0000000 0.8164966 1.0000000
```

2. Quelle est la formule de la dissimilarité de Jaccard telle que calculée dans la fonction dist.binary ? (regarder l'aide de cette fonction).

Correction. La dissimilarité de Jaccard calculée par R est égale à

$$\delta(e_i, e_j) = \sqrt{1 - \frac{a_{ij}}{a_{ij} + b_{\bar{i}j} + b_{i\bar{j}}}}$$

avec

$$\begin{aligned} a_{ij} &= Card\{p \in \{1, \ldots, P\}, X_i^p = X_j^p = 1\}, \\ b_{\bar{i}j} &= Card\{p \in \{1, \ldots, P\}, X_i^p = 0, X_j^p = 1\}, \\ b_{i\bar{j}} &= Card\{p \in \{1, \ldots, P\}, X_i^p = 1, X_j^p = 0\}. \end{aligned}$$

3. Recalculer sans utiliser R la dissimilarité de Jaccard entre e_1 et e_2 afin de retrouver la valeur calculée avec R (expliquer le calcul).

Correction. On a $X_1 = (1, 1, 0, 1)$ et $X_2 = (0, 0, 1, 0)$, donc $a_{ij} = 0$, $b_{\bar{i}j} = 1$ et $b_{i\bar{j}} = 3$. Par conséquent,

$$\delta(e_1, e_1) = \sqrt{1 - \frac{0}{0 + 1 + 3}} = 1$$

4. Effectuer sans utiliser R la classification ascendante hiérarchique obtenue à partir des dissimilarités de Jaccard et en utilisant la stratégie du maximum, en détaillant les étapes. Tracer le dendrogramme obtenu.

Correction.

— partition initiale : $\{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\}\}$

— étape 1 : on regroupe $\{e_3\}$ et $\{e_4\}$ car $\delta(e_3, e_4) = \mathbf{0.58}$ est la dissimilarité minimale. On obtient donc la partition $\{\{e_1\}, \{e_2\}, \{e_3, e_4\}, \{e_5\}\}$

— étape 2 : on a

$$\begin{aligned} \Delta(\{e_1\}, \{e_2\}) &= \delta(e_1, e_2) = 1, \\ \Delta(\{e_1\}, \{e_3, e_4\}) &= \max(\delta(e_1, e_3), \delta(e_1, e_4)) = 0.87, \\ \Delta(\{e_1\}, \{e_5\}) &= \delta(e_1, e_5) = 0.82, \\ \Delta(\{e_2\}, \{e_3, e_4\}) &= \max(\delta(e_2, e_3), \delta(e_2, e_4)) = 0.82, \\ \Delta(\{e_2\}, \{e_5\}) &= \delta(e_2, e_5) = 1, \\ \Delta(\{e_3, e_4\}, \{e_5\}) &= \max(\delta(e_3, e_5), \delta(e_4, e_5)) = 1, \end{aligned}$$

donc on a le choix entre regrouper $\{e_1\}$ et $\{e_5\}$ ou $\{e_2\}$ et $\{e_3, e_4\}$ (valeur minimale du critère : $\mathbf{0.82}$). Choissons $\{e_1\}$ et $\{e_5\}$. On obtient donc la partition $\{\{e_1, e_5\}, \{e_2\}, \{e_3, e_4\}\}$.

— étape 3 : on a

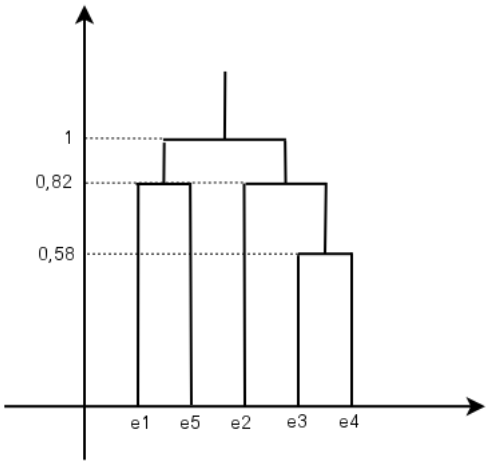
$$\begin{aligned} \Delta(\{e_1, e_5\}, \{e_2\}) &= \max(\delta(e_1, e_2), \delta(e_5, e_2)) = 1, \\ \Delta(\{e_1, e_5\}, \{e_3, e_4\}) &= \max(\Delta(\{e_1\}, \{e_3, e_4\}), \Delta(\{e_5\}, \{e_3, e_4\})) = 1, \\ \Delta(\{e_2\}, \{e_3, e_4\}) &= 0.82, \end{aligned}$$

donc on regroupe $\{e_2\}$ et $\{e_3, e_4\}$ (valeur minimale du critère : **0.82**). On obtient donc la partition $\{\{e_1, e_5\}, \{e_2, e_3, e_4\}\}$.

— étape 4 : on regroupe forcément $\{e_1, e_5\}$ et $\{e_2, e_3, e_4\}$. La valeur du critère est

$$\Delta(\{e_1, e_5\}, \{e_2, e_3, e_4\}) = \max(\Delta(\{e_1, e_5\}, \{e_2\}), \Delta(\{e_1, e_5\}, \{e_3, e_4\})) = \mathbf{1}$$

Le dendrogramme correspondant est :



5. Vérifier les calculs précédents en effectuant la classification hiérarchique à l'aide de R et afficher le dendrogramme. Retrouve-t-on nécessairement exactement les mêmes étapes et le même dendrogramme en procédant à la main ou avec R ? Si non, les différences ont-elles une importance ?

Correction.

```
cah = hclust(D,method="complete")
cah
cah$merge
plot(cah, hang=-1)
```

On obtient pour cah$merge :

```
     [,1] [,2]
[1,]   -3   -4
[2,]   -1   -5
[3,]   -2    1
[4,]    2    3
```

ce qui signifie qu'on a le même ordre de regroupement que celui obtenu à la question précédente : e_1 est regroupé avec e_5 d'abord, puis e_2 avec $\{e_3, e_4\}$. Ceci dit on aurait pu aussi bien choisir l'inverse et donc on aurait eu des étapes de CAH différentes. Le choix n'a pas d'importance puisque les valeurs du critère sont les mêmes. Ceci se voit sur le dendrogramme : on aurait exactement le même dendrogramme, même si les étapes étaient inversées.

6. Quel nombre de classes retient-on si l'on suit le critère du saut maximal ? Quelle partition cela donne-t-il ?

Correction. La saut maximal du critère est obtenu entre les valeurs 0.58 et 0.82. Donc on retient la partition en 4 classes :

$$\{\{e_1\}, \{e_2\}, \{e_3, e_4\}, \{e_5\}\}$$

Dans la suite, on suppose que l'on dispose en plus d'une autre variable Y binaire pour les 5 observations, telle que $Y_1 = Y_5 = 1$ et $Y_2 = Y_3 = Y_4 = 0$. On cherche à présent à construire un classifieur pour estimer Y à partir des X^i.

7. On considère un nouvel individu e_6 tel que $X_6^1 = X_6^2 = 1$ et $X_6^3 = X_6^4 = 0$. A l'aide de R, calculer les dissimilarités de e_6 avec les autres observations.

Correction.

```
Xnew = data.frame(X1=c(1,0,1,0,1,1),X2=c(1,0,0,0,0,1),
                  X3=c(0,1,1,1,0,0), X4=c(1,0,1,1,0,0))
dist.binary(Xnew,method=1)
```

renvoie

```
          1         2         3         4         5
2 1.0000000
3 0.7071068 0.8164966
4 0.8660254 0.7071068 0.5773503
5 0.8164966 1.0000000 0.8164966 1.0000000
6 0.5773503 1.0000000 0.8660254 1.0000000 0.7071068
```

On a donc

$$\delta(e_6, e_1) = 0.58, \quad \delta(e_6, e_2) = 1, \quad \delta(e_6, e_3) = 0.87, \quad \delta(e_6, e_4) = 1, \quad \delta(e_6, e_5) = 0.71.$$

8. Déterminer sans utiliser R la classe estimée de e_6 suivant la méthode des k-plus proches voisins, successivement pour $k = 1$, $k = 3$ et $k = 5$. (N.B. s'il y a des cas d'égalité on affectera aléatoirement).

Correction.

- pour $k = 1$: le plus proche voisin de e_6 est e_1, donc $\hat{Y}_6^{1-NN} = Y_1 = 1$.
- pour $k = 3$: les trois plus proches voisins de e_6 sont e_1, e_3 et $e5$. On a $Y_1 = Y_5 = 1$ et $Y_3 = 0$, donc la classe 1 est majoritaire. Par conséquent $\hat{Y}_6^{3-NN} = 1$.
- pour $k = 5$: ici les 5 plus proches voisins de e_6 sont tous les individus de la base (e_1 à e_5). Par conséquent on affecte e_6 dans la classe majoritaire. On a 3 individus de classe 0 et 2 de classe 1, donc $\hat{Y}_6^{5-NN} = 0$.

9. Toujours sans utiliser R, déterminer, successivement pour $k = 1$, $k = 3$ et $k = 5$, la classe estimée suivant la méthode des k-plus proches voisins, cette fois pour les individus e_1 à e_5, et calculer l'erreur apparente des trois classifieurs.

Correction.

— pour $k = 1$: le plus proche voisin de chaque individu de la base d'entraînement est forcément lui-même, donc on affecte chaque individu à sa propre classe. Ainsi

$$\hat{Y}_1^{1-NN} = Y_1 = 1, \quad \hat{Y}_2^{1-NN} = Y_2 = 0, \quad \hat{Y}_3^{1-NN} = Y_3 = 0,$$

$$\hat{Y}_4^{1-NN} = Y_4 = 0, \quad \hat{Y}_5^{1-NN} = Y_5 = 1,$$

et l'erreur apparente du classifieur $1 - NN$ est nulle.

— pour $k = 3$: ici on a :

— pour e_1 : les trois plus proches voisins de e_1 sont e_1, e_3 et e_5. On a $Y_1 = Y_5 = 1$ et $Y_3 = 0$, donc la classe 1 est majoritaire. Ainsi $Y_1^{3-NN} = 1$.

— pour e_2 : les trois plus proches voisins de e_2 sont e_2, e_3 et e_4, tous de classe 0. Ainsi $Y_2^{3-NN} = 0$.

— pour e_3 : les trois plus proches voisins de e_3 sont e_3, e_4 et e_1. On a $Y_1 = 1$ et $Y_3 = Y_4 = 0$, donc la classe 0 est majoritaire. Ainsi $Y_3^{3-NN} = 0$.

— pour e_4 : les trois plus proches voisins de e_4 sont e_4, e_3 et e_2, tous de classe 0. Ainsi $Y_4^{3-NN} = 0$.

— pour e_5 : les trois plus proches voisins de e_5 sont e_5, e_1 et e_3. On a $Y_1 = Y_5 = 1$ et $Y_3 = 0$, donc la classe 1 est majoritaire. Ainsi $Y_5^{3-NN} = 1$.

— pour $k = 5$: ici les 5 plus proches voisins de n'importe quel individu sont tous les individus de la base (e_1 à e_5). Par conséquent on affecte toujours dans la classe majoritaire. On a 3 individus de classe 0 et 2 de classe 1, donc

$$\hat{Y}_1^{5-NN} = \hat{Y}_2^{5-NN} = \hat{Y}_3^{5-NN} = \hat{Y}_4^{5-NN} = \hat{Y}_5^{5-NN} = 0.$$

10. Aurait-on pu utiliser la fonction `knn` de R pour utiliser la méthode des K-plus proches voisins sur ces données ? Pourquoi ?

Correction. Non car la fonction `knn` de R suppose que les données sont quantitatives et utilise la distance euclidienne pour calculer les dissimilarités.

Exercice 2
Dans cet exercice on va utiliser des données sur les passagers du paquebot Titanic, ayant coulé en 1912.

1. Charger les données `titanic_imputed` du package `DALEX`. Consulter la documentation sur ces données, puis répondez aux questions suivantes :
   — Que contiennent ces données ?
   — Quelle est la différence entre les données sur le Titanic dans ce package `DALEX` et celles du package `stablelearner` ?
   — Quelle est la différences entre les données `titanic_imputed` et les données `titanic` ?

Correction.

```
library(DALEX)
```

```
?titanic_imputed
```

Le tableau titanic_imputed contient des informations sur les passagers du paquebot Titanic : sexe, age, type de passager (1ere, 2e ou 3e classe, ou membre d'équipage, etc.), lieu d'embarquement, prix du ticket, nombre de frères et soeurs ou époux, , nombre de parents, ainsi qu'une dernière variable binaire spécifiant si le passager a suvécu ou non au naufrage. Il y a 2207 observations (passagers) et 8 variables (voir remarque ci-dessous) ; certaines variables sont qualitatives et d'autres numériques.

D'après la documentation, ces données sur le Titanic sont issues à l'origine du package `stablelearner`, avec quelques modifications mineures, principalement certaines modalités de variables qualitatives ont été renommées.

Par rapport aux données titanic, dans les données titanic_imputed, les valeurs manquantes sont remplacées sytématiquement par des valeurs estimées (imputation), afin que ces données puissent être utilisées par les méthodes d'analyse de données standards, qui souvent ne peuvent pas traiter le cas des données manquantes.

remarque : une autre différence importante entre titanic *et* titanic_imputed *est le fait que la variable "pays d'origine" est absente dans les données* titanic_imputed*. Ceci n'est curieusement pas mentionné dans la documentation !*

2. Séparer les données en une base d'entraînement contenant 2000 observations et une base de test contenant 207 observations.

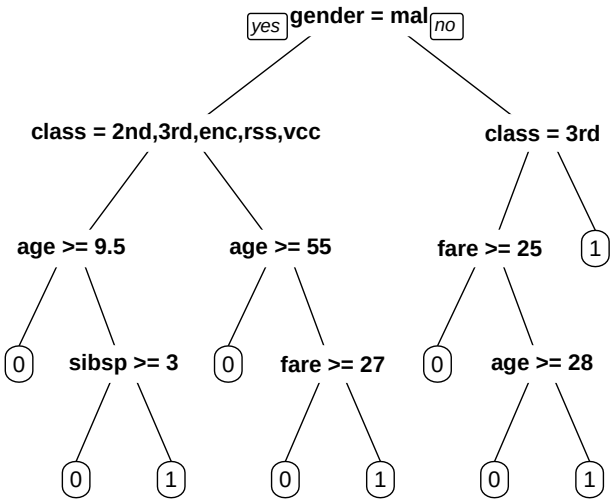
Correction.

```
n = 2207
ntest = 207
ind_test = sample(1:n,ntest)
data_train = titanic_imputed[-ind_test,]
data_test = titanic_imputed[ind_test,]
```

3. Utiliser la méthode CART, avec les paramètres par défaut, pour classer les données d'entraînement. Afficher l'arbre de décision obtenu et l'enregistrer dans un fichier PDF (Bouton "Export", puis "Save as PDF")

Correction.

```
library(rpart)
arbre = rpart(survived~., data_train)
library(rpart.plot)
prp(arbre)
```



4. Interpréter l'arbre de décision : quelles remarques "intuitives" sur les facteurs impliquant la survie d'un passager peut-on retrouver en observant l'arbre de décision obtenu ?

Correction. *remarque : ici le but avec cette question est de voir si l'on est capable de correctement lire et interpréter un arbre de décision. On formule donc des conjectures pour l'interprétation, comme dans cette correction, mais le fait que ces hypothèses soient pertinentes ou pas d'un point de vue historique est hors du cadre de cet examen et ne compte pas dans l'évaluation.*

On peut remarquer tout d'abord que deux variables ne sont pas utilisées dans l'arbre de décision : le lieu d'embarquement tout d'abord, ce qui semble logique : il n'y a a priori pas de raison que le lieu d'embarquement ait un lien avec le fait qu'un passager ait survécu ou non au naufrage. L'autre variable non utilisée est le nombre de parents ou enfants à bord. Ici c'est moins clair : on peut penser que cette variable est assez corrélée

avec l'age et que l'age du passager est une variable bien plus déterminante, mais on pourrait aussi argumenter dans l'autre sens (les passagers voyageant seuls avaient plus de chance de se sauver car plus mobiles, ou au contraire étaient moins prioritaires pour embarquer dans les canots de sauvetage).

On peut remarquer également que le sexe (homme/femme) est la variable la plus déterminante pour la classification vis à vis de la variable "survived" puisqu'il s'agit de la première question posée par l'arbre. On sait que les femmes (et enfants) étaient prioritaires pour embarquer dans les canots de sauvetages, ce qui peut être une explication.

Parmi les autres remarques possibles : si le passager est une femme de première ou seconde classe, l'arbre décide de le classer comme survivant. Ceci peut s'expliquer à nouveau par le fait que les femmes étaient prioritaires pour être embarquées dans les canots, ainsi que les passagers des meilleures classes. Autre remarque : plus le passager est jeune et plus il semble classé dans le groupe des survivants ; en effet trois noeuds dans l'arbre posent une questions sur l'âge du passager, et à chaque fois la branche correspondant aux plus âgés se retrouve classé en "non survivant". Là aussi on peut expliquer ça par la priorité accordée aux plus jeunes (enfants) pour entrer dans les canots, et éventuellement le désavantage physique lié à la vieillesse.

5. Calculer le taux d'erreurs obtenu avec l'arbre de décision précédent sur les données de test.

Correction.

```
cl_cart = predict(arbre, data_test[,1:7], type="class")
mean(cl_cart!=data_test$survived)
```

On obtient $20,3\%$ d'erreur.

6. Appliquer à présent la méthode des forêts aléatoires sur les données d'entraînement (toujours avec les paramètres par défaut). Calculer le taux d'erreurs obtenu sur les données test et commenter le résultat en le comparant avec celui de la question précédente.

Correction.

```
library(randomForest)
# ici on doit convertir la variable "survived" en facteur, car sinon la fonction
# randomForest fait de la régression.
data_train$survived = as.factor(data_train$survived)
rf = randomForest(survived~., data_train)
cl_rf = predict(rf, data_test[,1:7], type="class")
mean(cl_rf!=data_test$survived)
```

On obtient $19,8\%$ d'erreur. La méthode des forêts aléatoires est ici à peine plus performante que CART, mais il faudrait effectuer plus d'essais de validation (avec d'autres

séparations aléatoires en données train et test) pour conclure.

Dans la partie suivante, on va s'intéresser à une méthode permettant de quantifier l'importance d'une variable dans un classifieur, appelée méthode de permutation. L'idée est la suivante : on effectue une permutation aléatoire des valeurs de cette variable sur les données test, en laissant le reste des données de test inchangées. On applique ensuite le classifieur sur ces données modifiées et on calcule le taux d'erreurs E_{mod}. Enfin, le score d'importance de la variable est calculé par $E_{mod} - E_{org}$ où E_{org} est le taux d'erreurs original, c'est-à-dire calculé sur les données test sans permutation.

7. Pourquoi cette méthode permet à votre avis de bien quantifier l'importance d'une variable ? Pourquoi ne pourrait-on pas simplement supprimer la variable du tableau de données ?

Correction. Cette méthode semble cohérente pour évaluer l'importance d'une variable. En effet lorsque les valeurs d'une variable sont permutées dans un tableau de données, on obtient pour cette variable des valeurs complètement incohérentes. Par conséquent si le variable est importante dans le classifieur, la performance du classifieur doit être fortement diminuée par la permutation. A l'opposé, si par exemple la variable n'est pas utilisée dans le classifieur (comme c'était le cas pour le lieu d'embarquement pour l'arbre obtenu par CART), la permutation n'aura aucun impact sur la classification.

Le fait de supprimer la variable du tableau semble aussi une bonne idée pour évaluer son importance dans la classification. Mais on ne peut pas le faire si le but est, comme indiqué, d'évaluer l'importance de la variable "dans un classifieur" : en effet si le classifieur est donné à l'avance, il faut forcément lui donner en entrée un tableau contenant les mêmes variables que le tableau d'origine. Si on supprime la variable, alors on est obligé de calculer un nouveau classifieur sur le tableau de données sans la variable. On ne serait plus alors en train d'évaluer un classifieur, mais d'évaluer une méthode de classification, ce qui est un peu différent.

8. Appliquer la méthode pour le classifieur obtenu précédemment avec la méthode CART, pour la variable "gender" : calculer le tableau de données test modifié, puis le taux d'erreur sur les données test et enfin le score.

Correction.

```
data_test_permut = data_test
data_test_permut$gender = sample(data_test_permut$gender)
cl_cart_permut = predict(arbre, data_test_permut[,1:7], type="class")
E_mod = mean(cl_cart_permut!=data_test$survived)
E_mod
E_org = mean(cl_cart!=data_test$survived)
score = E_mod - E_org
score
```

On obtient 32.9% pour E_{mod} et un score de 12.6%.

9. A présent faire une boucle sur l'ensemble des variables afin de calculer le score d'importance de chaque variable, toujours sur le classifieur CART. Commenter les résultats : d'après ces scores, quelles variables sont déterminantes pour la survie des passagers du Titanic, et quelles variables sont peu importantes ?

Correction.

```
score = rep(0,7)
for(k in 1:7)
{
  data_test_permut = data_test
  data_test_permut[,k] = sample(data_test_permut[,k])
  cl_rf_permut = predict(rf, data_test_permut[,1:7], type="class")
  E_mod = mean(cl_rf_permut!=data_test$survived)
  E_org = mean(cl_rf!=data_test$survived)
  score[k] = E_mod - E_org
}
score
```

On obtient

```
[1]  0.19806763  0.01449275  0.15942029  0.00000000
          -0.01449275  0.00000000  0.00000000
```

Quatre variables ont un score d'importance nul ou négatif : le lieu d'embarcation, le nombre d'enfants/parents, le nombre de frères/sœurs/époux et le prix du ticket. Pour les deux premières variables c'est logique puisque ces deux variables ne sont pas utilisées dans l'arbre. Pour les deux autres il faudrait examiner plus précisément les données pour comprendre, mais on peut penser que les nœuds relatifs à ces variables dans l'arbre ne concernent qu'un faible nombre d'observations.

La variable "age" a aussi très peu d'importance. Au final, d'après ces scores, seules les variables "gender" et "class" semblent déterminantes pour ce classifieur, les autres n'ont presque pas d'importance.

10. Comme le score dépend d'une permutation aléatoire, on peut affiner la mesure en faisant une moyenne sur plusieurs tests. Modifier le code de la question précédente pour obtenir des scores moyennés sur 100 permutations. Les conclusions de la questions précédentes sont-elles modifiées ?

Correction.

```
score = rep(0,7)
for(k in 1:7)
{
  for(i in 1:100)
  {
    data_test_permut = data_test
    data_test_permut[,k] = sample(data_test_permut[,k])
```

```
    cl_cart_permut = predict(arbre, data_test_permut[,1:7], type="class")
    E_mod = mean(cl_cart_permut!=data_test$survived)
    score[k] = score[k] + E_mod - E_org
  }
  score[k] = score[k] / 100
}
score
```

On obtient

```
[1] 1.537198e-01 1.657005e-02 1.320290e-01 0.000000e+00
       6.956522e-03 4.830918e-05 0.000000e+00
```

soit en arrondissant à deux décimales (commande round(score,2)) :

```
[1] 0.15 0.02 0.13 0.00 0.01 0.00 0.00
```

Ces résultats confirment les conclusions précédentes : pour ce classifieur, seules les variables "gender" et "class" sont importantes, les autres n'ont pas ou presque pas d'importance.

11. Enfin effectuer la même analyse en remplaçant le classifieur CART par le classifieur obtenu par la méthode des forêts aléatoires. Commenter les résultats.

Correction.

```
score_rf = rep(0,7)
E_org_rf = mean(cl_rf!=data_test$survived)
for(k in 1:7)
{
  for(i in 1:100)
  {
    data_test_permut = data_test
    data_test_permut[,k] = sample(data_test_permut[,k])
    cl_rf_permut = predict(rf, data_test_permut[,1:7], type="class")
    E_mod_rf = mean(cl_rf_permut!=data_test$survived)
    score_rf[k] = score_rf[k] + E_mod_rf - E_org_rf
  }
  score_rf[k] = score_rf[k] / 100
}
round(score_rf,2)
```

On obtient

```
[1]  0.18  0.00  0.05  0.00  0.00 -0.01  0.00
```

Ici la conclusion est un peu différente : les seules variables importantes sont toujours "gender" et "class", mais "gender" est beaucoup plus importante que "class" pour le classifieur obtenu avec les forêts aléatoires. Cependant on ne peut pas vraiment tirer une

conclusion de cette différence par rapport à CART, vu que les performances des deux classifieurs semblent proches.