EISTI	**ING2-MI** **RATTRAPAGE DE DATAMINING 2018-2019**
Durée : 3h	Exammanager

Consignes

Vous devez rendre sur Exammanager

- ✓ un document **pdf** contenant :
  - Les lignes de codes utilisées
  - Les graphiques obtenus
  - Les réponses aux questions
  - Et le cas échéant les screenshot permettant de justifier vos réponses.
- ✓ Votre programme R commenté

N'oubliez pas de mettre **votre nom** en entête de tous les documents.

Si besoin, vous pouvez rendre une copie papier si vous ne pouvez pas répondre à certaines questions dans le document pdf.

Le jeu de données utilisé dans cet examen provient du package mlbench.

```
library(mlbench)
data(Ozone)
help(Ozone)
```

Los Angeles ozone pollution data, 1976 : A data frame with 366 observations on 13 variables, each observation is one day.

L'objectif est de prédire la quantité journalière d'ozone (V4) en fonction des autres variables.

1. Préparation des données

La variable Ozone est quantitative, il s'agit donc d'un problème de régression.

```
hist(Ozone$V4)
```

Afin de le transformer en problème de classification, nous allons découper la variable ozone (V4) en 2 classes : Normal et Elevé, suivant les seuils 0-30, >30. Dans un premier temps, on supprime les lignes pour lesquelles la variable Ozone n'est pas renseignée :

```
Mydata=Ozone[-c(144,190,204,264,267),]
Classes=cut(Mydata$V4,c(0,20,max(Mydata$V4)),labels = c("Normal","Eleve"))
Mydata[,4]=Classes
```

1.1. Faites une représentation graphique appropriée pour la variable Classes. Que constatez-vous ?

1.2. A partir de la matrice de confusion suivante

		Prédiction	
		Normal (+)	Elevé (-)
Observé	Normal (+)	TP	FN
	Elevé (-)	FP	TN

Le critère d'erreur classique est

$$t_e = \frac{FN+FP}{TP+FN+FP+TN}.$$

Nous proposons d'utiliser le critère d'erreur suivant

$$t_m = \frac{1}{2}\left[\frac{FN}{TP+FN}+\frac{FP}{FP+TN}\right].$$

a) Pouvez-vous expliquer pourquoi ?

b) On considère un classifieur qui prédit toujours la classe Normal. Quels sont les taux d'erreur classique, t_e, et modifié, t_m, sur la base Mydata ?

1.3. Quelles variables explicatives (attributs) sont quantitatives ? Qualitatives ? S'il y a des variables qualitatives ne sont pas détectées comme « factor », alors il faut les transformer avec l'instruction `as.factor`.

Ensuite, on remplace toutes les valeurs manquantes des variables explicatives numériques par leur médiane :

```
for (i in 5:13)
{
  variable=Mydata[,i]
  mediane=median(variable,na.rm=TRUE)
  index.na=which(is.na(variable))
  variable[index.na]=mediane
  Mydata[,i]=variable
}
```

1.4. Quelles méthodes de prévision peuvent prendre en compte des variables explicatives quantitatives et qualitatives en même temps : Naive Bayes, LDA, QDA, Arbre de décision, Réseau de neurones, Forêt aléatoire ?

Normaliser toutes les variables quantitatives entre [-1,1]

```
min=apply(Mydata[,5:13],2,min)
max=apply(Mydata[,5:13],2,max)
data.scaled = scale(Mydata[,5:13], center = min, scale = max-min) # Scale the data
Mydata [,5:13]=data.scaled
```

1.5. Construire un base d'apprentissage et une base test avec un ratio 2/3 – 1/3.
Que devez-vous vérifier pour ces bases ?

2. Réseau de neurones

Première partie
Dans un premier temps, nous allons considérer un réseau de neurones simple avec uniquement les variables explicatives V5 et V6 et 2 neurones sur la couche cachée. Les fonctions d'activation entre la couche d'entrée et la couche cachée et entre la couche cachée et la couche de sortie est la fonction sigmoide ($logit^{-1}$).

2.1. Combien de poids y-a-t-il à ajuster dans ce réseau ? Justifier votre réponse.

2.2. Ajuster ce réseau de neurones avec la fonction `nnet` avec comme base `Mydata`, en précisant dans les paramètres : `Wts=rep(0,9)`.

2.3. Quelle est la conséquence de l'instruction `Wts=rep(0,9)` ?

2.4. Combien faut-il d'itérations avec que l'algorithme converge ? Quelle est alors la valeur de l'erreur quadratique moyenne ?

2.5. Sur votre copie, dessinez le réseau de neurones en précisant les poids obtenus sur chaque arrête.

2.6. Donner l'expression mathématique des deux neurones H_1 et H_2 de la couche cachée, en fonction de V_4, V_5 et des poids calculés par l'algorithme.

2.5. Donner l'expression mathématique du neurone de sortie O_1 en fonction de H_1 et H_2 et des poids calculés par l'algorithme. A quelle probabilité correspond O_1 ?

2.6. A partir des formules établies aux questions 5 et 6, calculer les valeurs de H_1 et H_2 pour une nouvelle entrée telle que V_5=1 et V_6=0. En déduire la sortie du réseau de neurones.

Deuxième partie
Nous allons maintenant voir s'il est possible de prédire la quantité d'ozone uniquement avec les jours de la semaine (V3).

2.7. Ajuster un réseau avec 2 neurones de V4 en fonction de V3 avec comme base `Mydata` et en précisant `Wts=rep(0,17)`.

2.8. Combien y-a-t-il de neurones sur la couche d'entrée. A quoi correspondent-ils ?

2.9. Est-ce que certains jours ont plus d'influence sur la sortie du réseau que d'autres ?

Troisième partie
Nous allons maintenant construire le modèle complet.

2.10. Ajuster un réseau avec 5 neurones sur la couche cachée, toutes les variables explicatives en utilisant la base d'apprentissage.

2.11. Répéter l'ajustement du réseau plusieurs fois jusqu'à obtenir la meilleure erreur quadratique moyenne. Quelle est cette valeur ? D'où vient l'aléatoire dans l'algorithme ?

2.12. Construire la matrice de confusion sur la base test. En déduire le taux d'erreur de prévision classique et le taux de prévision modifié (cf. question 1.2.)

3. Foret aléatoire

Nous allons construire un foret aléatoire pour prédire la quantité d'ozone.

3.1. Construire un foret aléatoire sur la base d'apprentissage avec 20 arbres et 5 variables par arbre. Utiliser la fonction `randomForest`.

3.2. Construire la matrice de confusion sur la base test. En déduire le taux d'erreur de prévision classique et le taux de prévision modifié (cf. question 1.2.). Comparer avec le réseau de neurones.

3.3. Quelles variables ont été les plus importantes dans les split ? Faites une représentation graphique pour justifier votre réponse.

4. Questions diverses

4.1. Soit la variable cible suivante : [0,0,0,1,1,1,1,1]. Quelle est l'entropie de cette variable ?

4.2. Dans cet exercice, on considère les points de IR^2 suivants :

$M_1(2,0)$, $M_2(0,1)$,$M_3(0,2)$,$M_4(2,2)$, $M_5(7,0)$,$M_6(8,0)$,$M_7(0,8)$ et $M_8(8,8)$.

La distance entre les points est la distance de Manhattan,

$$d(x,y) = \sum_{i=1}^{d} |x_i - y_i|$$

et la distance entre les classes est la distance de Ward,

$$d_{Ward}(C_1, C_2) = \frac{n_1 \times n_2}{n_1 + n_2} d(g_1, g_2)$$

où g_i sont les centres de gravités des classes et n_i leur effectif.

a) Appliquez la méthode de k-means pour répartir les points en deux classes avec l'initialisation $G_1=M_7$ et $G_2=M_8$.

b) Calculez le pourcentage d'inertie expliqué par la partition obtenue.

c) Expliquez la différence entre la méthode des k-means et la classification ascendante hiérarchique.