Université Paris Ouest - Nanterre La Défense
Master2 ISEFAR Data Mining - Classification
CM et TP (Groupe 1) : Ana Karina FERMIN
Tp (Groupe 2) : Mélanie Zetlaoui

TP2 : Méthode des k plus proches voisins (k-ppv)

Exercice

1. Les données

   (a) Acquérir et visualiser sous R les données Iris (voir TP1)

   (b) Restreindre les données aux longueurs et aux largeurs des pétales pour deux types d'iris : versicolor et iris virginica.

   ```
   irismod=subset(iris,!(Species=="setosa"))
   D=irismod[,c("Petal.Length","Petal.Width","Species")]
   ```

2. Le sous-échantillon d'entrainement (Dtrain) et de test (Dtest).

   ```
   p=0.75; n=dim(D)[1]; ntrain=floor(n*p); ntest= n-ntrain
   ind=sample(1:n,n); ind_train=ind[1:ntrain]; ind_test=ind[(ntrain+1):n]
   Dtrain=D[ind_train,] ; Dtest=D[ind_test,]
   ```

   Tracer dans un même fenêtre l'ensemble de points dans le plan associées à Dtrain et Dtest.

3. Classifieur des k-plus proches voisins, estimation de l'erreur et frontière de décision

   (a) Charger la librairie class grâce à la commande `library(class)` puis évaluer le classifieur, associé à la méthode **5-ppv**, dans tous les points de test.

   ```
   predict.Dtest = knn(Dtrain[,c("Petal.Length","Petal.Width")],
   Dtest[,c("Petal.Length","Petal.Width")],Dtrain[,"Species"], k=5)
   ```

   (b) Donner la matrice de confusion et l'erreur sur l'échantillon de test à l'aide des commandes

   ```
   table(Dtest[,"Species"] , predict.Dtest)
   mean(Dtest[,"Species"]!= predict.Dtest)
   ```

   Combien de points ont été mal classés par votre classifieur ? Changer la valeur de k (par exemple k=1). Répéter l'item (a), recalculer la matrice de confusion et l'erreur sur l'échantillon de test . Que peut-on conclure ?

   (c) Donner le taux d'erreur par validation croisée à l'aide du code suivant

```
  K=3; k=5
  Error.knn=dim(K)
  ind_Fold=sample(rep(1:K,ceiling(n/K)),n)
  for (i in 1:K){
    ind_train=which(ind_Fold!=i); ind_test=which(ind_Fold==i)
    Dtrain=D[ind_train,]; Dtest=D[ind_test,]
    predict.Dtest=knn(Dtrain[,1:2],Dtest[,1:2], Dtrain[,3], k)
    Error.knn[i]=mean(Dtest[,3]!= predict.Dtest)
  }
mean(Error.knn)
```

(d) Utiliser les commandes suivantes pour construire une grille pour les axes du plan et représenter les frontières de décision données par la règle de `kppv`.

```
Length = seq(min(Dtrain[,"Petal.Length"]),max(Dtrain[,"Petal.Length"]),
 length=50)
Width = seq(min(Dtrain[,"Petal.Width"]),max(Dtrain[,"Petal.Width"]),
 length=50)
Grid=expand.grid(Petal.Length = Length, Petal.Width= Width)
pred.Grid = knn(Dtrain[,c("Petal.Length","Petal.Width")],
                     Grid[,c("Petal.Length","Petal.Width")],
                     Dtrain[,"Species"],k=5)

contour(Length,Width,matrix(as.numeric(droplevels(pred.Grid)),50),
        level=1.5, labels="knn", xlab="Longueur du pétale",
     ylab="Largeur du pétale",main = "Frontière de décision (5-ppv)")
```

Qu'observe-t-on dans le graphique ? Pourquoi dans l'argument de la fonction `contour` on a choisi `level = 1.5` ?

Ajouter les points d'entrainement dans le graphique précédente.

Refaire un autre graphique avec les points d'entrainement et la frontière de décision pour `k=1`. Comparer les deux graphiques.

4. **Sélection du bon `k` par K-validation croisée**

   Fixer `K=3` et créer un vecteur des valeurs de `k`, par exemple `k = 1:20`. Écrire un boucle qui permet de calculer, pour chaque valeur de `k`, le taux d'erreur par K-Fold CV. Choisir la(les) valeur(s) de `k` qui produise (produisent) la plus petite erreur.

5. **Le meilleur classifieur parmi la famille considérée**

   Évaluer le classifieur, associé à la méthode **k-ppv** avec un des "bons" `k` choissi dans la partie précédente, dans tous les points de test. Tracer la frontière de décision et l'ensemble de points d'entrainement.