

TP5

Exercice 1

KPPV (K plus proche voisin) sous Weka

1. Ouvrez WEKA et charger le fichier « weather.nominal.arff »
2. Soit l'exemple de test suivant : test (sunny, cool, high, TRUE, ?). Remplir le tableau ci-dessous. La distance est le nombre des valeurs différentes :

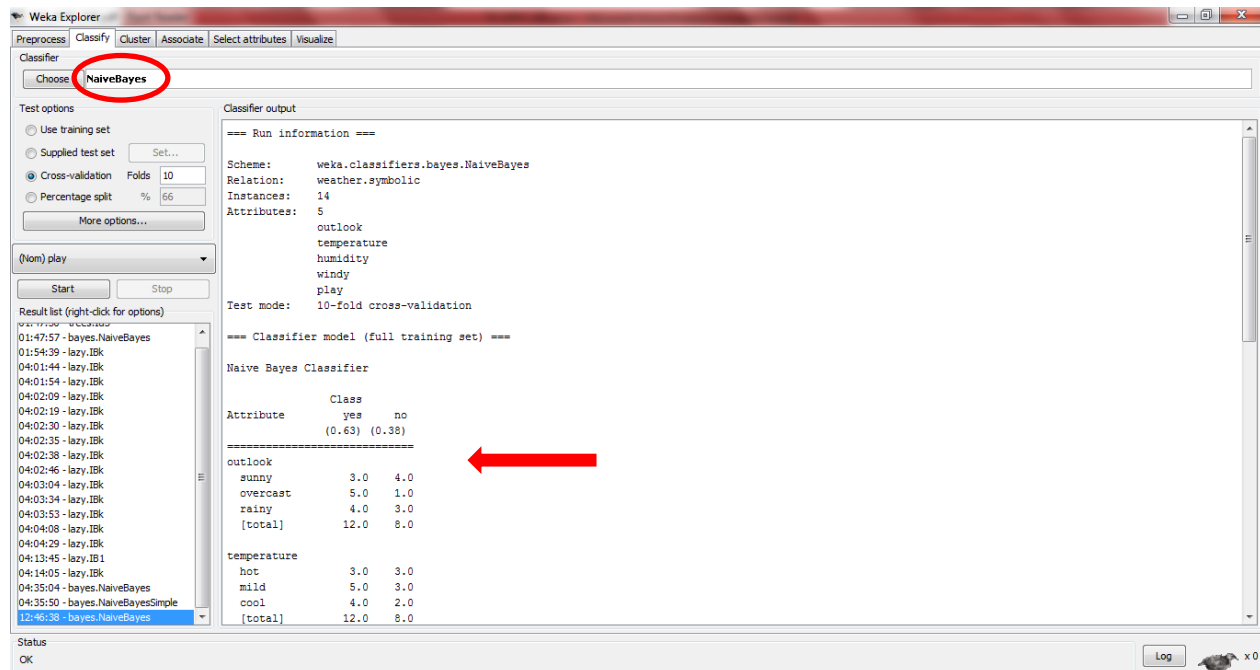
Tuple	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Play														
Distance (test, tuple)														

3. K-PPV : affecter l'exemple de test à la classe du plus proche voisin de la classe majoritaire parmi K voisins.
 - a. K=1 -> ?
 - b. K=3 -> ?
 - c. K=5 -> ?
 - d. K=14 -> ? (ZeroR)
4. Que peut-on remarquer ?
5. Proposez une solution possible à ce problème
6. Créez le fichier correspondant à l'exemple précédant et testez le résultat trouvé avec Weka
Pour lancer KPPV sous Weka, il faut d'abord être sous l'onglet « classify » puis choisir dans « classifier » le classifieur « **IBk** » dans le répertoire des classifieur « lazy ».
7. Testez les différentes options existantes : nombre K, fonctions de distance, ...

Exercice 2

Naïve bayes sous Weka

1. Pourquoi certains classifieurs de Bayes sont dits « naïfs » ?
2. Ouvrez WEKA et charger le fichier « weather.nominal.arff »
3. Lancez Weka et testez la classification avec naïve bayes (NB) sous Weka.
Pour lancer NB sous Weka, il faut d'abord être sous l'onglet « classify » puis choisir dans « classifier » le classifieur « **NaiveBayes** » dans le répertoire des classifieur « bayes ».



4. Observez et analysez le modèle de classification et la distribution des attributs dans la partie « Classifier model »
5. Comparez le temps d'exécution de Naive Bayes avec KPPV
6. Expliquez cette différence.

Naïve bayes sous R

1. Ouvrez R et installez le package « *e1071* » avec

```
install.packages("e1071")
```
2. Chargez le jeu de données

```
birth <- read.csv("iris.arff.csv")
summary(birth)
```
3. Chargez la bibliothèque avec

```
library(e1071)
```
4. Construire le modèle de classification

```
modele <- naiveBayes(low ~ ., data = birth)
print(modele)
```
5. Vérifiez la prédiction du modèle construit sur les données d'apprentissage

```
pred <- predict(modele, newdata = birth[,2:7])
```
6. Affichez la matrice de confusion et le taux d'erreur

```
mc <- table(birth$low, pred)
print(mc)
error <- (mc[1,2]+mc[2,1])/sum(mc)
print(error)
```