

TP : Arbre de décision

Exercice 1 :

- 1- Installer le package **tree**
- 2- Charger les données Iris
- 3- Créer l'arbre avec la fonction **tree**.
- 4- Afficher les informations de l'arbre avec la fonction **summary**.
- 5- Afficher l'arbre avec la fonction **plot**.
- 6- Afficher les points des différents dans le plan des paramètres (Petal.Width, Sepal.Width).
C'est-à-dire les abscisses = Petal.Width et les ordonnées = Sepal.Width).
- 7- Utiliser la fonction **partition.tree** pour partitionner les données de l'arbre créé dans la question 3.

Exercice 2 :

- 1- Charger, exécuter et Commenter le script **Arbre_Tenis.R**.

Exercice 3 :

Récupérer le jeu de données d'apprentissage **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.

```
rm(list=ls())  
data <- read.table(file="../synth_train.txt", header=TRUE)  
dim(data)  
## [1] 100 3  
data$y <- as.factor(data$y)
```
- 2- Charger le package rpart (**rpart** = recursive **partitioning**) et consulter l'aide de la fonction rpart.

```
library(rpart)  
help(rpart)
```
- 3- Construire un arbre de classification t à l'aide de la fonction **rpart** (attention la fonction demande une formule du style y ~. et des données data=data_synth). Faire afficher l'arbre en tapant t, tracer l'arbre à l'aide des fonctions plot(t) puis text(t).
- 4- Calculer l'erreur d'apprentissage du prédicteur obtenu.
- 5- Charger le jeu de données test puis calculer le taux d'erreur test.
- 6- Faire de même avec l'arbre maximal (on pourra regarder la fonction **rpart.control** qui permet de régler les règles de construction d'un arbre).