TP Erreur de Resubstitution

1. Analyse de l'arbre et application de la fonction de perte

L'arbre de décision ci-dessus est construit avec un paramètre `nsplit=2`, ce qui signifie qu'il a été divisé deux fois pour créer des nœuds supplémentaires.

Combien de fois doit-on appliquer la fonction de perte ? La fonction de perte est appliquée à chaque feuille de l'arbre pour évaluer l'erreur de classification. Ici, il y a quatre feuilles, donc la fonction de perte s'applique quatre fois.

Explication de la première application La première application de la fonction de perte se fait en analysant la première division de l'arbre, qui sépare les données en fonction de la variable `Temps`.

- Si `Temps = Ensl, Plvx`, alors la classification est `Oui` (5/9), avec 5 bonnes classifications et 4 erreurs.
- Si `Temps = Cvrt`, alors la classification est `Oui` (0/4), donc pas d'erreur ici.

2. Un autre exemple d'application avec un résultat différent

Si l'on modifie légèrement la structure de l'arbre en introduisant une nouvelle variable de décision (ex. `Vent`), cela pourrait donner une répartition différente des erreurs et affecter la classification des données.

3. Calcul de l'erreur de resubstitution

L'erreur de resubstitution est calculée en comptant le nombre total de mauvaises classifications sur le nombre total d'observations.

Dans cet arbre :

- Le nœud `Temps = Ensl, Plvx` contient 9 observations, dont 4 mal classées.
- Le nœud `Temps = Cvrt` contient 4 observations, toutes bien classées.
- Le nœud `Humidité = Elev` contient 5 observations, toutes bien classées.
- Le nœud `Humidité = Nrml` contient 4 observations, dont 1 mal classée.

Total des erreurs : **4 + 0 + 0 + 1 = 5** Total des observations : **9 + 4 = 13**

L'erreur de resubstitution est donc : 513≈0.38\frac{5}{13} \approx 0.38

4. Explication de la realerror (0.4)

La realerror est obtenue en ajustant l'erreur de resubstitution à l'aide d'un facteur de pénalisation qui prend en compte la complexité de l'arbre et les données utilisées pour l'entraînement.

Une légère augmentation de l'erreur estimée peut être due à l'optimisme biaisé du modèle qui a été ajusté sur l'ensemble d'entraînement sans validation croisée.

Ainsi, la realerror de **0.4** reflète une estimation plus réaliste de la capacité généralisante du modèle sur des données non vues.