

HAX907X TP2 : Arbres

```
# Construction des classifieur
dt_entropy = tree.DecisionTreeClassifier(criterion='entropy')
dt_gini = tree.DecisionTreeClassifier(criterion='gini')

# Simulation de l'échantillon
n = 456
data = rand_checkers(n1=n//4, n2=n//4, n3=n//4, n4=n//4)
n_samples = len(data)
X = data[:, :2]
Y = np.asarray(data[:, -1], dtype=int)

# Entraînement des deux modèles
dt_gini.fit(X, Y)
dt_entropy.fit(X, Y)

print("Gini criterion")
print(dt_gini.score(X, Y))

print("Entropy criterion")
print(dt_entropy.score(X, Y))
```

```
Gini criterion
1.0
Entropy criterion
1.0
```

```
dt_entropy.max_depth = np.argmax(1-scores_entropy)+1
plt.figure(figsize=(6,3.2))
frontiere(lambda x: dt_entropy.predict(x.reshape((1, -1))), X, Y, step=100)
plt.draw()
```

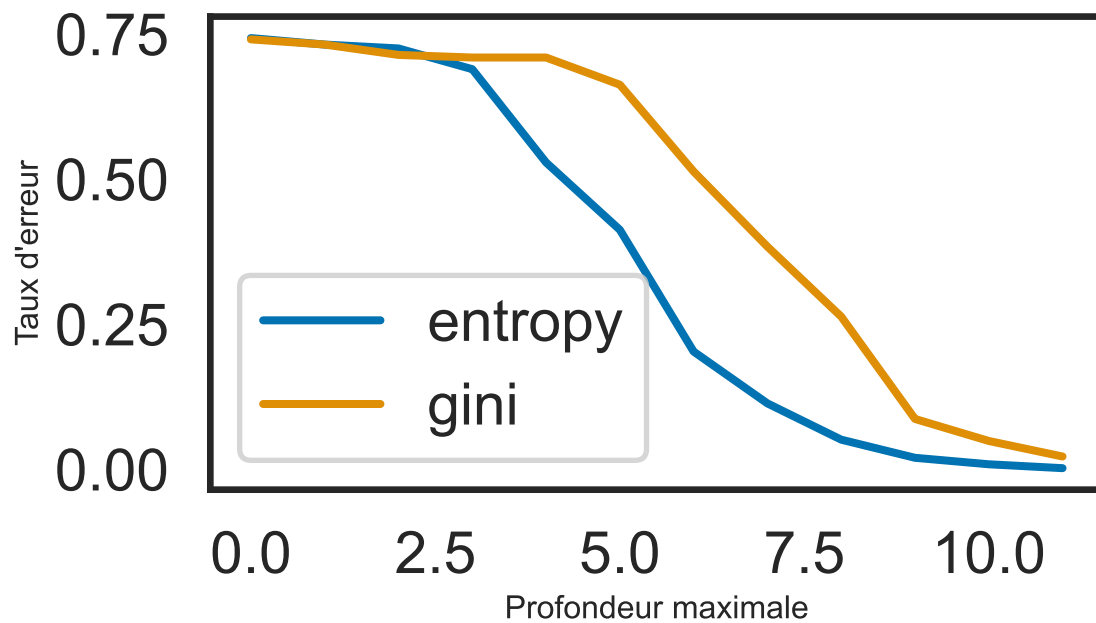


Figure 1: Pourcentage d'erreurs commises en fonction de la profondeur maximale de l'arbre

```
print("Best scores with entropy criterion: ", dt_entropy.score(X, Y))
```

Best scores with entropy criterion: 0.9977678571428571

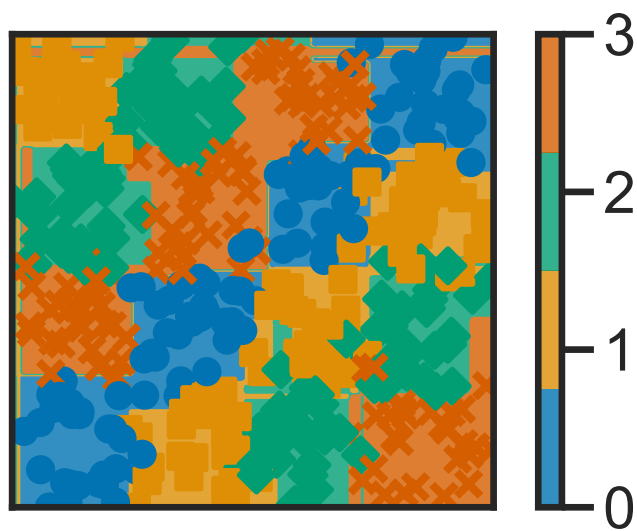


Figure 2: Frontières pour la meilleur profondeur (entropie)