demonstration_dt

November 10, 2025

0.0.1 Ch6. Apprentissage Supervisé – Algorithme arbre de décision

Démo simple (intuition)

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor,␣
 ↪plot_tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, mean_squared_error, r2_score
from sklearn.datasets import fetch_california_housing
import seaborn as sns
```

[ ]:
```
X= [
    [1, 1, 20],  # Tanjia=1, Humour=1, Age=20
    [1, 0, 25],
    [0, 1, 28],
    [1, 1, 32],
    [0, 1, 35],
    [1, 0, 45],
    [0, 0, 55],
    [1, 1, 58]
]
y= [1, 0, 1, 1, 1, 0, 0, 0]  # 1 = Marrakechi
```

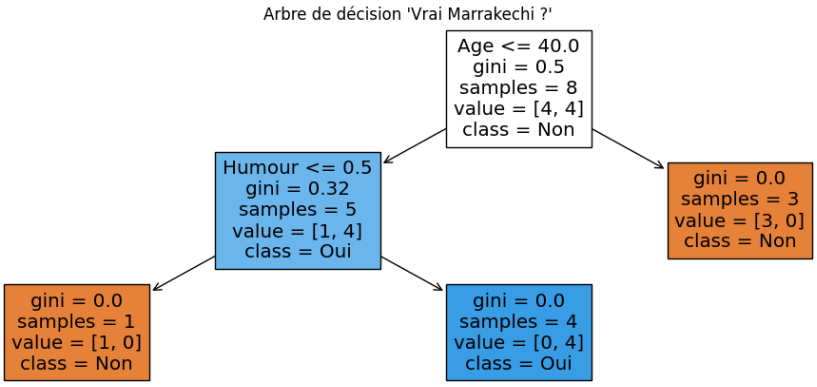
[5]:
```
clf = DecisionTreeClassifier(criterion='gini', max_depth=3)
```

[6]:
```
clf.fit(X, y)
```

[6]:
```
DecisionTreeClassifier(max_depth=3)
```

[43]:
```
plt.figure(figsize=(15,5))
plot_tree(clf,
          feature_names=['Tanjia','Humour','Age'],
          class_names=['Non','Oui'],
          filled=True
          )
plt.title("Arbre de décision 'Vrai Marrakechi ?'")
```

```
plt.show()
```



Classification (Heart Disease)

[19]:
```
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer()
X_cancer = data.data
y_cancer = data.target
```

[20]:
```
X_train, X_test, y_train, y_test = train_test_split(X_cancer,
                                                    y_cancer,
                                                    test_size=0.3,
                                                    random_state=42)
clf_cancer = DecisionTreeClassifier(criterion='gini',
                                    max_depth=5,
                                    random_state=42)
clf_cancer.fit(X_train, y_train)
y_pred = clf_cancer.predict(X_test)
print(classification_report(y_test, y_pred))
```

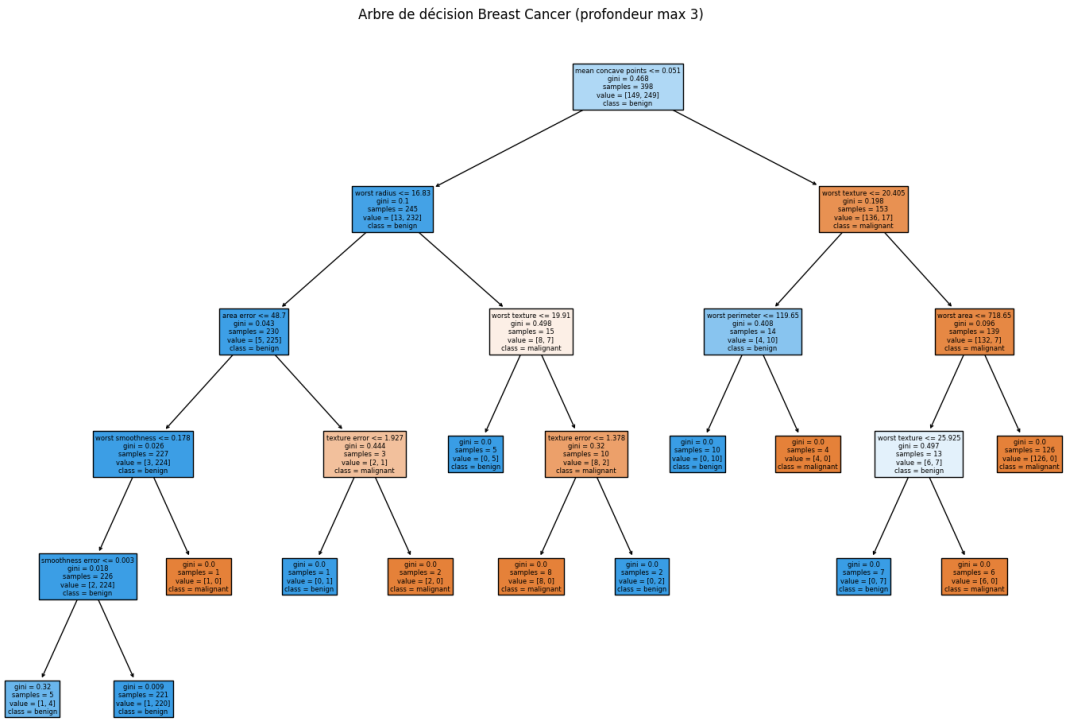
```
              precision    recall  f1-score   support

           0       0.94      0.94      0.94        63
           1       0.96      0.96      0.96       108

    accuracy                           0.95       171
   macro avg       0.95      0.95      0.95       171
weighted avg       0.95      0.95      0.95       171
```

[ ]:
```
plt.figure(figsize=(18,12))
plot_tree(clf_cancer,
          filled=True,
          max_depth=5,
```

```
          feature_names=data.feature_names,
          class_names=data.target_names)
plt.title("Arbre de décision Breast Cancer (profondeur max 3)")
plt.show()
```



Régression (California Housing)

[34]:
```
X_reg, y_reg = fetch_california_housing(return_X_y=True)
X_train_reg, X_test_reg, y_train_reg, y_test_reg = train_test_split(X_reg,␣
 ↪y_reg,
                                                                    test_size=0.
 ↪3,
                                                                    ␣
 ↪random_state=42)

reg = DecisionTreeRegressor(max_depth=5, random_state=42)
reg.fit(X_train_reg, y_train_reg)
```

[34]:
```
DecisionTreeRegressor(max_depth=5, random_state=42)
```

[38]:
```
y_pred = reg.predict(X_test_reg)
print("Arbre de Régression California Housing")
print(f"MSE : {mean_squared_error(y_test_reg, y_pred)*100:.2f}%")
```

```
print(f"R²  : {r2_score(y_test_reg, y_pred)*100:.2f}%")
```

```
Arbre de Régression California Housing
MSE : 52.11%
R²  : 60.30%
```

[48]:
```
# Visualisation (arbre réduit)
plt.figure(figsize=(16,6))
plot_tree(reg, filled=True, max_depth=3,␣
 ↪feature_names=fetch_california_housing().feature_names)
plt.title("Arbre de régression California Housing (profondeur max 3)")
plt.show()
```