Arbre de Décision - CART

Arbre de Décision (CART)

Théorie

L'algorithme CART (**Classification and Regression Trees**) est un modèle d'apprentissage supervisé qui construit un arbre de décision en divisant l'espace des caractéristiques en sous-ensembles homogènes.

Évaluation des performances

Lorsqu'on évalue un modèle de classification, plusieurs métriques sont utilisées :

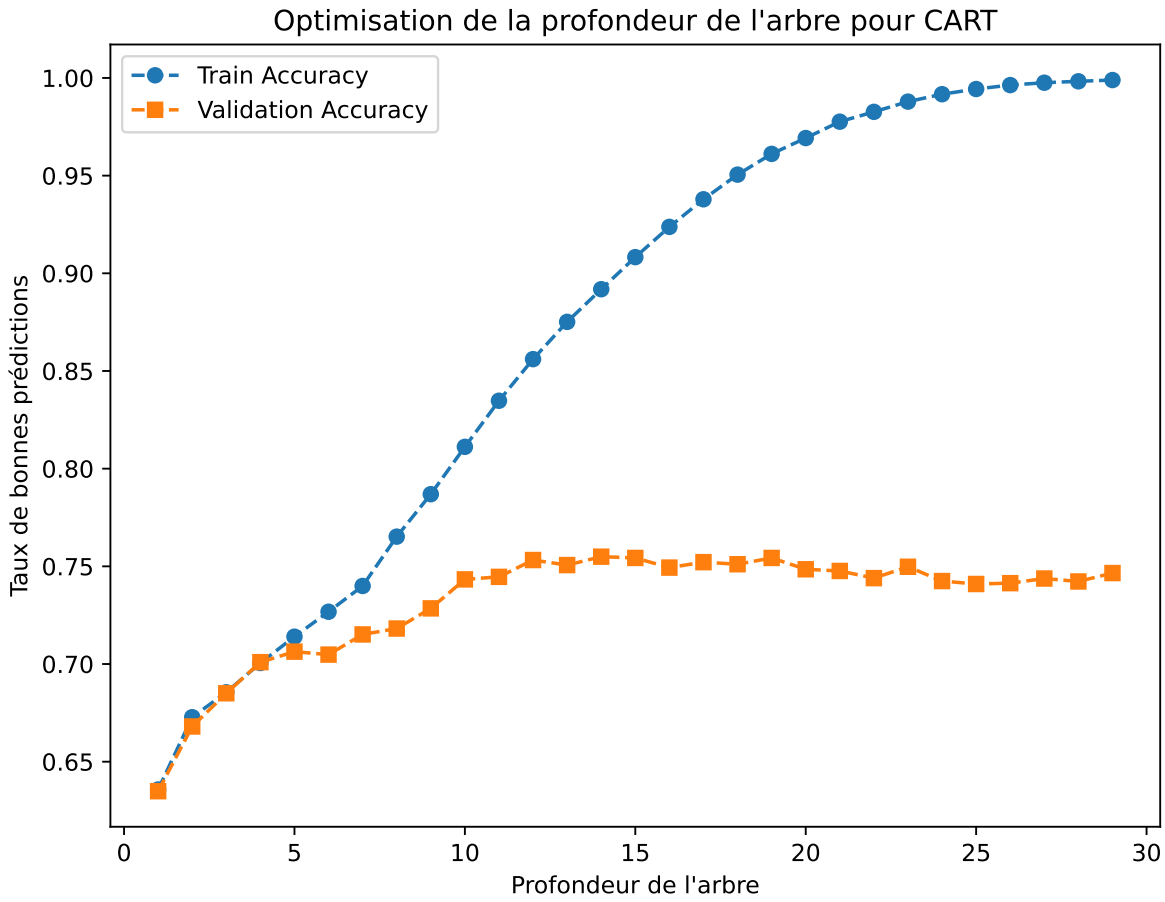
- **Matrice de confusion** : Tableau qui résume les performances du modèle en comparant les vraies classes aux classes prédites. Les lignes correspondent aux **classes réelles**, et les colonnes aux **classes prédites**.
- **Accuracy (Précision globale)** : Proportion des prédictions correctes parmi l'ensemble des données.
- **Precision (Précision par classe)** : Nombre de vrais positifs divisé par la somme des vrais positifs et des faux positifs. Indique la fiabilité des prédictions positives.
- **Recall (Rappel)** : Nombre de vrais positifs divisé par la somme des vrais positifs et des faux négatifs. Indique la capacité du modèle à détecter les échantillons positifs.
- **F1-score** : Moyenne harmonique entre précision et rappel, utile lorsque les classes sont déséquilibrées.

Exemple en Python

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.model_selection import train_test_split

# Chargement des ensembles de données déjà préparés
train_data = pd.read_csv('covertype_train.csv')
val_data = pd.read_csv('covertype_val.csv')
test_data = pd.read_csv('covertype_test.csv')

# Préparation des données
X_train = train_data.drop('Cover_Type', axis=1)
y_train = train_data['Cover_Type']

X_val = val_data.drop('Cover_Type', axis=1)
y_val = val_data['Cover_Type']

X_test = test_data.drop('Cover_Type', axis=1)
y_test = test_data['Cover_Type']

# Recherche de la meilleure profondeur d'arbre en utilisant l'ensemble de validation
depth_range = range(1, 30)
train_accuracies = []
val_accuracies = []

for depth in depth_range:
    tree = DecisionTreeClassifier(max_depth=depth, random_state=42)
    tree.fit(X_train, y_train)

    # Évaluation sur l'ensemble d'entraînement
    y_train_pred = tree.predict(X_train)
    train_accuracies.append(accuracy_score(y_train, y_train_pred))

    # Évaluation sur l'ensemble de validation
    y_val_pred = tree.predict(X_val)
    val_accuracies.append(accuracy_score(y_val, y_val_pred))

# Sélection de la meilleure profondeur basée sur l'ensemble de validation
best_depth = depth_range[val_accuracies.index(max(val_accuracies))]
print(f"Meilleure profondeur d'arbre: {best_depth}")
```

```
# Affichage du graphique comparant l'entraînement et la validation
plt.figure(figsize=(8, 6))
plt.plot(depth_range, train_accuracies, marker='o', linestyle='dashed', label='Train Accuracy
plt.plot(depth_range, val_accuracies, marker='s', linestyle='dashed', label='Validation Accu
plt.xlabel("Profondeur de l'arbre")
plt.ylabel("Taux de bonnes prédictions")
plt.title("Optimisation de la profondeur de l'arbre pour CART")
plt.legend()
plt.show()

# Modèle final avec la meilleure profondeur
tree = DecisionTreeClassifier(max_depth=best_depth, random_state=42)
tree.fit(X_train, y_train)

# Évaluation sur l'ensemble de test
y_test_pred = tree.predict(X_test)

# Affichage de la matrice de confusion avec annotations
conf_matrix = confusion_matrix(y_test, y_test_pred)
print("\nMatrice de confusion (les lignes représentent les vraies classes et les colonnes les
print(conf_matrix)

print("\nÉvaluation sur l'ensemble de test")
print(classification_report(y_test, y_test_pred))
```

```
Meilleure profondeur d'arbre: 14
```




```
Matrice de confusion (les lignes représentent les vraies classes et les colonnes les classes
[[1281  391    0    0    3    0   33]
 [ 399 1802   25    0   13   21    1]
 [   2   34  205    5    0   35    0]
 [   0    0    5   12    0    4    0]
 [   2   50    1    0   21    0    0]
 [   1   29   52    0    1   61    0]
 [  39    3    0    0    0    0  117]]

Évaluation sur l'ensemble de test
              precision    recall  f1-score   support

           1       0.74      0.75      0.75      1708
           2       0.78      0.80      0.79      2261
           3       0.71      0.73      0.72       281
```

4	0.71	0.57	0.63	21
5	0.55	0.28	0.38	74
6	0.50	0.42	0.46	144
7	0.77	0.74	0.75	159
accuracy			0.75	4648
macro avg	0.68	0.61	0.64	4648
weighted avg	0.75	0.75	0.75	4648