

Travaux Dirigés N°7 : Apprentissage supervisé—arbres de décision

Objectif du TP :

- Construire un arbre de décision pour classer une iris
- Utilisation d'algorithme et de python et la librairie Sickit-Learn
- Rédiger le compte rendu d'expériences

Réalisation du TP :

Le travail réalisé au cours de ce TP porte sur la classification des fleurs du dataset Iris en utilisant un arbre de décision. L'objectif principal est d'explorer le processus de création d'un modèle, son entraînement, son évaluation et la visualisation de l'arbre résultant.

Étapes du TP :

1.Chargement des données :

- Le jeu de données Iris est chargé à l'aide de la fonction `'load_iris()'` de scikit-learn.

2. Préparation des données

- Les caractéristiques et les étiquettes sont stockées dans des DataFrames (`'donnees'` et `'target'`).
- Les noms des colonnes dans le DataFrame `'donnees'` sont définis pour faciliter la compréhension

3. Division des données :

- Les données sont divisées en ensembles d'entraînement (`'x_train'`, `'y_train'`) et de test (`'x_test'`, `'y_test'`) en utilisant `'train_test_split'`. 70% des données sont utilisées pour l'entraînement, et 30% pour le test. La division des données est une pratique courante en apprentissage automatique pour évaluer la performance d'un modèle sur des données qu'il n'a pas vues pendant l'entraînement.

4. Création et entraînement du modèle :

- Un modèle d'arbre de décision est créé avec certains paramètres spécifiés (critère "entropy", profondeur maximale 3, et minimum 5 échantillons par feuille).

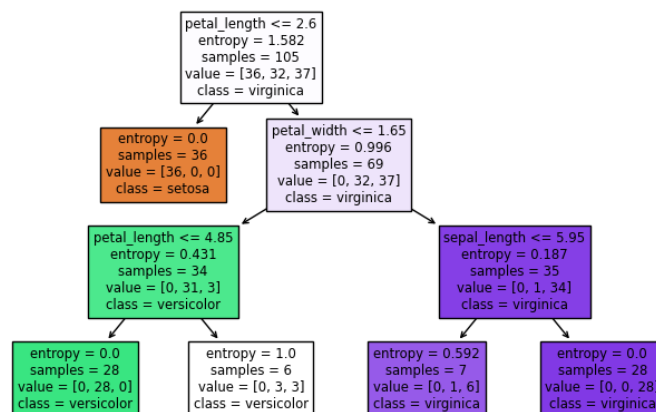
- Le modèle est ensuite entraîné sur les données d'entraînement à l'aide de la méthode `fit`.

5. Évaluation du modèle :

- Les prédictions sont faites sur les données de test.
- La précision du modèle est calculée à l'aide de `accuracy_score`.
- La matrice de confusion, le rapport de classification, l'erreur absolue moyenne, l'erreur quadratique moyenne et l'erreur quadratique moyenne racine sont affichés pour évaluer les performances du modèle.

6. Visualisation de l'arbre de décision avec **paramètres et criterion= entropy**:

- L'arbre de décision résultant est visualisé à l'aide de `plot_tree`.



```

DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_leaf=5,
                        random_state=100)
95.55555555555556
[[14  0  0]
 [ 0 17  1]
 [ 0  1 12]]
      precision    recall  f1-score   support

     0       1.00      1.00      1.00        14
     1       0.94      0.94      0.94        18
     2       0.92      0.92      0.92        13

   accuracy       0.96
  macro avg       0.96
weighted avg       0.96

Mean Absolute Error: 0.044444444444444446
Mean Squared Error: 0.044444444444444446
Root Mean Squared Error: 0.21081851067789195

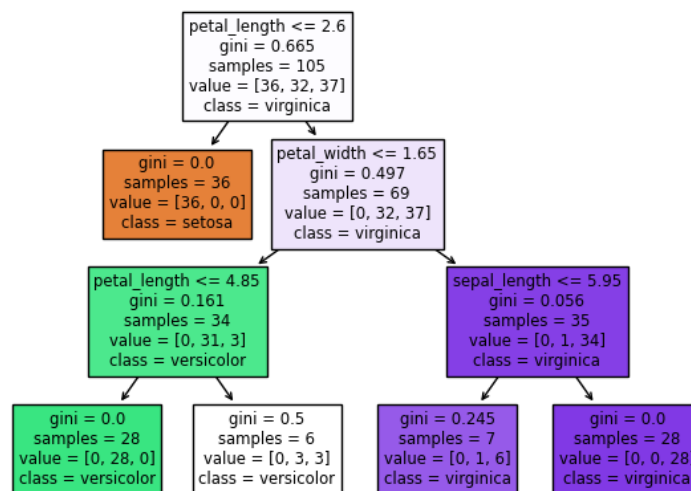
```

Observations :

- Le modèle d'arbre de décision est entraîné avec le critère d'entropie, qui mesure l'incertitude d'une distribution.

- La profondeur maximale de l'arbre est limitée à 3 niveaux pour éviter le surajustement.

6. Visualisation de l'arbre de décision avec **paramètres et criterion= gini**:



```
DecisionTreeClassifier(max_depth=3, min_samples_leaf=5, random_state=100)
95.55555555555556
[[14  0  0]
 [ 0 17  1]
 [ 0  1 12]]
      precision    recall  f1-score   support

     0       1.00      1.00      1.00        14
     1       0.94      0.94      0.94        18
     2       0.92      0.92      0.92        13

   accuracy       0.96
  macro avg       0.96
weighted avg       0.96

Mean Absolute Error: 0.044444444444444446
Mean Squared Error: 0.044444444444444446
Root Mean Squared Error: 0.21081851067789195
```

On remarque que le résultat est le même en utilisant entropy et gini. Cela est dû à la simplicité du problème et au jeu de données ainsi que des paramètres identiques.

Voici une dernière comparaison entre gini et entropy:

Accuracy (Gini): 0.9555555555555556						
Confusion Matrix (Gini):						
[[14 0 0]						
[0 17 1]						
[0 1 12]]						
Classification Report (Gini):						
		precision	recall	f1-score	support	
	0	1.00	1.00	1.00	14	
	1	0.94	0.94	0.94	18	
	2	0.92	0.92	0.92	13	
	accuracy			0.96	45	
	macro avg	0.96	0.96	0.96	45	
	weighted avg	0.96	0.96	0.96	45	

Accuracy (Entropy): 0.9555555555555556						
Confusion Matrix (Entropy):						
[[14 0 0]						
[0 17 1]						
[0 1 12]]						
Classification Report (Entropy):						
		precision	recall	f1-score	support	
	0	1.00	1.00	1.00	14	
	1	0.94	0.94	0.94	18	
	2	0.92	0.92	0.92	13	
	accuracy			0.96	45	
	macro avg	0.96	0.96	0.96	45	
	weighted avg	0.96	0.96	0.96	45	

Decision Tree - Gini



Decision Tree - Entropy



(Disponible dans le fichier zippé)

Conclusion du TP :

En résumé, le programme a créé, entraîné et évalué un modèle d'arbre de décision en utilisant les critères de Gini et Entropy pour classer les fleurs iris. Les données ont été divisées en ensembles d'entraînement et de test, la performance du modèle a été

évaluée, et l'arbre de décision résultant a été visualisé. Cette introduction pratique met en lumière l'utilisation des arbres de décision pour des tâches de classification.