TP3 : Les arbres de décision

Classification

Objectifs

Dans ce TP, nous allons tout d'abord étudier l'algorithme des arbres de décision en classification.

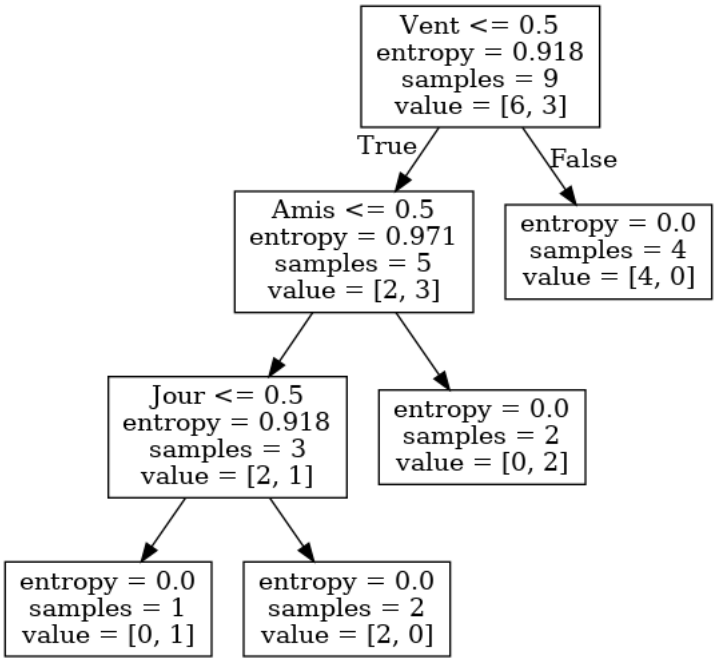
Un premier exemple

Dans cette partie nous allons voir un premier exemple sur la base de données barbecue vue en TD.

Etudiez le code suivant :

```
import pandas as pd
from sklearn import tree


def main():
    data = pd.read_csv("data/barbecue.csv")
    print(data)
    print(data['barbecue'])
    x_train = data
    y_train = data['barbecue']
    del x_train['barbecue']

    classifier = tree.DecisionTreeClassifier(criterion='entropy')
    classifier.fit(x_train, y_train)

    tree.export_graphviz(classifier, out_file='tree.dot',
                         feature_names=['Meteo', 'Amis', 'Vent', 'Jour'])


if __name__ == '__main__':
    main()
```

Ce code donne :



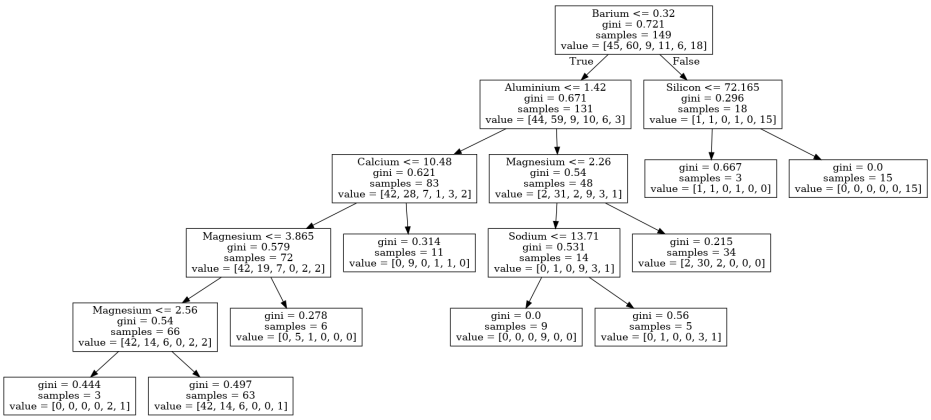
La base `glass`

Nous allons travailler sur la base de données glass. Il s'agit d'apprendre le type d'un verre (batiment, voiture, ...) en fonction de différentes caractéristiques chimiques.

A partir du TP précédent et de l'exemple fournit précédemment, écrivez un modèle d'arbre de décision pour cette base de données. Attention à ne pas négliger la partie d'analyse des données.

Exemple de résultat attendu :

```
Train score: 0.7785234899328859, Test score 0.7076923076923077
[[20  5  0  0  0  0]
 [ 1 13  0  2  0  0]
 [ 6  2  0  0  0  0]
 [ 0  0  0  2  0  0]
 [ 0  1  0  0  2  0]
 [ 2  0  0  0  0  9]]
```



Regression

Objectif

Les arbres de décision peuvent aussi être utilisés pour des problèmes de regression. Dans cette seconde partie, nous allons étudier la base de données winequality-red. Il s'agit de prédire la qualité d'un vin en fonction de différentes caractéristiques chimiques. La dernière colonne correspond à la qualité du vin, représentée par une note entière. Vous pouvez également vous amuser avec la base sur le vin blanc.

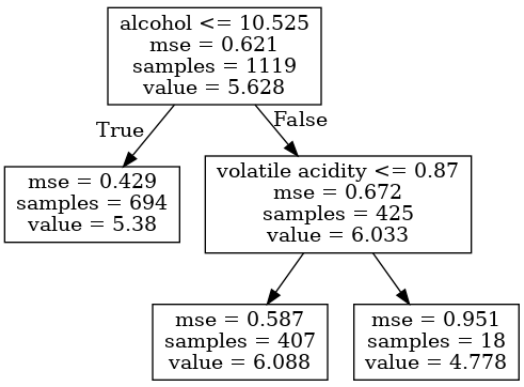
A vous de jouer

Comme pour la première partie, récupérez la base, analysez les données, et effectuez un apprentissage avec les arbres de décision.
Vous devez utiliser les bibliothèques suivantes :

- `from sklearn.neighbors import DecisionTreeRegressor`
- `from sklearn.model_selection import train_test_split`
- `from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score`

Exemple de résultat attendu :

```
Coefficient of determination: 0.265941458484
MAE: 0.5815440506
MSE: 0.530038694448
```



Résumé

- Les arbres de décision sont des algorithmes simples à mettre en place et faciles à comprendre.
- Ils souffrent toutefois d'un fort surapprentissage possible, il est donc important de limiter la taille des arbres.