Approches interprétables "by design"

Exercice 1 Mesures de discrimination

Dans cet exercice, on étudie les 3 mesures de discrimination présentées en cours (entropie de Shannon H_S, indice de diversité de Gini H_G et mesure d'ambiguïté H_Y). On considère le cas simple où $p(v_1) = 1$ et $\forall j \neq 1,\ p(v_j) = 0$ et où il n'y a que deux classes c_1 et c_2. On note p la probabilité $p(c_1)$.

1. Tracer, sur la même figure, les courbes donnant H en fonction de p pour chaque mesure.
2. En tirant aléatoirement p pour constituer la distribution $(p, 1-p)$, il est possible de comparer les valeurs pour chaque mesure de discrimination pour cette distribution. En réalisant plusieurs tirages aléatoires pour p, tracer les courbes affichant, pour tout couple de mesures (H_1, H_2) la valeur de H_1 en fonction de la valeur de H_2.
3. En étudiant les résultats obtenus pour les deux questions précédentes, que peut-on déduire sur les différences entre ces mesures ?

Exercice 2 Arbres de décision en présence de données symboliques

1. En utilisant la fonction `tree.DecisionTreeClassifier` de la librairie `scikit-learn`, construire (en utilisant l'entropie de Shannon) et tester un arbre de décision sur les données `digits`.
2. Récupérer le fichier archive data.tgz sur le Moodle. Il contient, dans le répertoire date, un ensemble de jeux de données à utiliser. Après avoir appliqué l'approche *one hot encoding*, et en utilisant la fonction `tree.DecisionTreeClassifier`, construire l'arbre de décision pour la base `elections.csv` fournie Moodle.
3. Implémenter l'algorithme de construction d'arbres de décision en présence de données symboliques (qui ne nécessite donc pas l'utilisation du *one hot encoding*). Pour cette implémentation, respecter la même API que pour `tree.DecisionTreeClassifier` en fournissant une fonction `fit` et une fonction `predict`. De plus, une fonction `predict_xai` doit donner la classe d'un exemple à classer et le chemin suivi dans l'arbre pour obtenir cette classe.
4. En utilisant la fonction `predict_xai` fournir l'explication de la classification d'un exemple. Comparer l'interprétabilité de la réponse par rapport à celles produites par les algorithmes précédents que vous avez vus.
5. Même question en utilisant les différentes bases *Mushrooms* fournies dans le fichier archive.

Annexes

Chargement de la base `load_digits` avec `scikit-learn`

```
from sklearn.datasets import load_digits
base_digits = load_digits()
print("Descriptions de la base digits:\n", base_digits.data)
```

One Hot Encoding `get_dummies` avec `pandas`

Arbres de décision avec `scikit-learn`

```
from sklearn import tree
# déclaration / initialisation d'un arbre
mon_arbre = tree.DecisionTreeClassifier(criterion='entropy')
```

```
# construction de l'arbre avec la base des Iris prise totalement
mon_arbre.fit(X,Y)
# classification avec l'arbre construit
mon_arbre.predict(test_data)
```