Université de Lorraine – M2 informatique - Spécialité IL+ MLFS + AVR

TP 2

Premiers pas avec Python & Scikit-Learn

Installation

Librairies utiles pour Python (>= 3.6):
- pandas
- numpy
- sklearn
- matplotlib

Le plus pratique est d'utiliser Jupyter Notebook
Le mieux (en attendant d'avoir un Jupyter hub à la FST) c'est d'installer Anaconda (https:// www.anaconda.com/products/individual), pour une installation facilitée de Jupyter et d'autres outils pour Python (et R). Il faut compter près de 5Go pour Anaconda…

Une fois Anaconda installé, ouvrir le client web pour Jupyter et créer un nouveau notebook avec (un noyau) Python. Le mien (sur MacOS) avait l'URL http://localhost:8888/tree

Partie 1 : Tutorial

Suivre le petit tutorial Oreilly sur Arche pour créer votre premier notebook (en y insérant les commentaires qui aident à comprendre les étapes et vous permettront de réutiliser les commandes) et le sauvegarder.

Partie 2 : Construction et visualisation d'arbres de décision et encodage de données

a) Charger les données breast cancer (`sklearn.datasets.load_breast_cancer`)

b) Tester et comprendre les différentes façons d'encoder les variables nominales en nombres :
   - `sklearn.preprocessing.LabelEncoder`
   - `sklearn.preprocessing.OrdinalEncoder`
   - `sklearn.preprocessing.LabelBinarizer`
   - `sklearn.preprocessing.OneHotEncoder`

   en les appliquant à un attribut de votre choix.

Choisir une manière d'encoder chaque attribut du jeu de données.

c) Construire un arbre de décision à partir des données préparées. Le visualiser.

d) Calculer la performance du modèle en utilisant la fonction score. La métrique par défaut est l'accuracy mais vous pouvez opter pour une autre métrique (`precision_score` ou `recall_score`)

e) Utiliser la validation croisée (folds = 10) pour évaluer le taux d'erreur (`model_selection.cross_val_score(model, x, y, cv = K)`.

f) Tester une recherche sur grille (grid search) pour sélectionner les meilleurs hyperparamètres : `sklearn.model_selection.GridSearchCV(dtree_model, param_grid, nfolds)`

   En donnant comme paramètres `param_grid = {'criterion':['gini','entropy'], 'max_depth': np.arange(3, 15)}` et `nfolds = 10`

   Visualiser le nouvel arbre de décision.