Lab04: Thực hành bài toán máy học phân lớp dựa trên cây quyết định (Decision Tree)

1. **Mục tiêu**:
   Am hiểu cách vận dụng thư viện sklearn mà cụ thể là các hàm trong thư viện tree của sklearn để thực hiện bài toán máy học đơn giản.
2. **Hướng dẫn cài đặt Python và Jupyter Notebook (hoặc Jupyter Lab) trên Windows 10/11**
   a. Cài đặt Python 3.x trên Windows: download python 3.x mới nhất trên website python.org rồi cài đặt Python.
   b. Thêm đường dẫn Python trên Windows vào biến môi trường (environment variables). Ví dụ: đường dẫn đến Python là:
   C:\Users\<username>\AppData\Local\Programs\Python\Python310\
   C:\Users\<username>\AppData\Local\Programs\Python\Python310\Scripts\
   c. Cài đặt Jupyter Notebook/Jupyter Lab: thực hiện 1 số lệnh sau in cmd.
   pip install --upgrade pip
   python -m pip install --upgrade pip
   pip install jupyter
   hoặc
   python -m pip install jupyter
   hoặc
   pip install jupyterlab
   d. Thực thi Jupyter Notebook: mở cmd, gõ lệnh:
   jupyter notebook
   hoặc
   jupyter lab
   e. Cài đặt các thư viện khác trong môi trường Python: *pip install <tên thư viện>*
3. **Tập dữ liệu Iris**
   Xét bài toán phân loại hoa IRIS dựa trên thông tin về kích thước của cánh hoa và đài hoa. Tập dữ liệu này có 150 phần tử, mỗi loại hoa có 50 phần tử. Dữ liệu có 4 thuộc tính (sepal length, sepal width, petal length, petal width) và 3 lớp (3 loại hoa Iris: Setosa, Versicolour, Virginica).
   Tập dữ liệu này có thể download từ trang UCI: *https://archive.ics.uci.edu/ml/datasets/iris* rồi đọc dữ liệu bằng lệnh *read_csv* của thư viện *Pandas* hoặc có thể nạp dữ liệu có sẵn bởi

thư viện Sklearn.

a. Sử dụng hàm train_test_split của thư viện cross_validation trong sklearn để tách dữ liệu chính thành các tập dữ liệu đào tạo (training) và tập kiểm tra (testing) X_train, Y_train, X_test, Y_test:
from sklearn.cross_validation import train_test_split
X_train, Y_train, X_test, Y_test = train_test_split(… test_size=1/3.0, random_states=5)
Xây dựng mô hình cây quyết định dựa trên chỉ số Gini với độ sâu của cây bằng 3, nút nhánh có ít nhất 5 phần tử.

4. Xây dựng mô hình phân loại với hàm DecisionTreeClassifier, dự đoán và tính độ chính xác

a. Sử dụng hàm DecisionTreeClassifier để tạo ra model phân loại (training) để thực hiện phân loại cho testing data:
from sklearn.tree import DecisionTreeClassifier
clf_gini = DecisionTreeClassifier(criterion = "gini", random_state = 100, max_depth=3, min_samples_leaf=5)
clf_gini.fit(X_train, y_train)

b. Dự đoán nhãn cho các phần tử trong tập kiểm tra: Tập kiểm tra (*X_test*) không tham gia vào việc training model. Sau khi đã training xong với các tập dữ liệu (*X_train, Y_train*), ta có thể để máy thực hiện dự đoán.
y_pred = clf_gini.predict(X_test)
clf_gini.predict([[4, 4, 3, 3]])

c. Tính độ chính xác cho giá trị dự đoán của phần tử trong tập kiểm tra:
from sklearn.metrics import accuracy_score
*print "Accuracy is ", accuracy_score(y_test,y_pred)*100*

d. Tính độ chính xác cho giá trị dự đoán thông qua ma trận con:
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, y_pred, labels=[2,0,1])

5. Xây dựng mô hình phân loại với hàm *DecisionTreeRegressor* (cây hồi quy), dự đoán và tính độ chính xác

Cũng thực hiện các bước xây dựng mô hình, dự đoán và tính độ chính xác giống câu trên nhưng sử dụng hàm *DecisionTreeRegressor* để xây dựng cây model.

Tham khảo thêm một số links:
https://www.kaggle.com/code/anivalogy/decision-trees-practice

https://www.kaggle.com/code/akashrajsrinivasan/decision-tree-a-detailed-overview

https://www.science.smith.edu/~jcrouser/SDS293/labs/lab14-py.html