Thực hành Máy Học ứng dụng
Buổi 3: Giải thuật cây quyết định

Mục tiêu:

- Củng cố lý thuyết và cài đặt giải thuật cây quyết định
- Kiểm thử và đánh giá theo nghi thức hold-out và K-fold

1. HƯỚNG DẪN THỰC HÀNH

A. Bài toán phân lớp – chỉ số Gini

- *Sử dụng tập dữ liệu có sẵn "iris"*

```
#Lay file iris truc tiep tu sklearn
from sklearn.datasets import load_iris
iris_dt = load_iris()
iris_dt.data[1:5] # thuoc tinh cua tap iris
iris_dt.target[1:5] #gia tri cua nhan /class
```

- *Phân chia tập dữ liệu để xây dựng mô hình và kiểm tra theo nghi thức Hold-out*

```
from sklearn.cross_validation import train_test_split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(iris_dt.data, iris_dt.target, test_size=1/3.0,
random_state=5)

X_train[1:6]
X_train[1:6,1:3]
y_train[1:6]
X_test[6:10]
y_test[6:10]
```

- *Xây dựng mô hình cây quyết định dựa trên chỉ số Gini với độ sâu của cây bằng 3, nút nhánh ít nhất có 5 phần tử.*

```
# Xay dung mo hinh cay quyet dinh dua tren chi so Gini
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)
```

- *Dự đoán nhãn cho các phần tử trong tập kiểm tra*

```
# du doan

y_pred = clf_gini.predict(X_test)
y_test
clf_gini.predict([[4, 4, 3, 3]])
```

- *Tính độ chính xác cho giá trị dự đoán của phần tử trong tập kiểm tra*

```
# tinh do chinh xac
from sklearn.metrics import accuracy_score
print ("Accuracy is ", accuracy_score(y_test,y_pred)*100)
```

Kết quả thu được

```
Accuracy is  94.0
```

- *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])
```

Kết quả thu được

```
>>> confusion_matrix(y_test, y_pred, labels=[2,0,1])
array([[15,  0,  2],
       [ 0, 16,  0],
       [ 1,  0, 16]])
```

B. Nghi thức K-fold

Thực hiện các bước tương tự ở phần A với nghi thức K-fold

class `sklearn.model_selection`
`KFold`(*n_splits=3*, *shuffle=False*, *random_state=None*)

- ***n_splits*** *: int, default=3*
  Number of folds. Must be at least 2.
- ***shuffle*** *: boolean, optional*
  Whether to shuffle the data before splitting into batches.
- ***random_state*** *: int, RandomState instance or None, optional, default=None*
  If int, random_state is the seed used by the random number generator; If RandomState instance, random_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random. Used when `shuffle` *== True.*

- Sử dụng nghi thức k-fold để phân chia tập dữ liệu "iris" với k=15 với hàm Kfold

```
from sklearn.model_selection import KFold
kf= KFold(n_splits=15) # chia tập dữ liệu thành 15 phần
```

```
for train_index, test_index in kf.split(X): # split(): Generate indices to split data into training and test set
    print("Train:", train_index, "Test:", test_index) # in giá trị chỉ số của tập huấn luyện và tập kiểm tra
    X_train, X_test = X[train_index], X[test_index] # tạo biến X_train và X_Test để lưu trữ thuộc tính của tập t
    y_train, y_test = y[train_index], y[test_index] # tạo biến y_train và y_Test để lưu trữ nhãn của tập train v
    print ("X_test", X_test)              # In thuộc tính của dữ liệu kiểm tra
    print("=======================")
```

Nếu dữ liệu đọc từ file thông qua thư viện Pandas thì truy xuất giá trị X, y thông qua code bên dưới (X.iloc[train_index,])

```
for train_index, test_index in kf.split(X):
     # print("TRAIN:", train_index, "TEST:", test_index)
    X_train, X_test = X.iloc[train_index,], X.iloc[test_index,]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]
#    print ("X_test", X_test)                # In thuộc tính của dữ liệu kiểm tra
    print("=======================")
```

C. Bài toán hồi quy – cây hồi quy

1. Cho tập dữ liệu housing_RT.csv có dạng:

	price	lotsize	bedrooms	bathrms	stories
1	38500.0	4000	2	1	1
2	49500.0	3060	3	1	1
3	60500.0	6650	3	1	2
4	61000.0	6360	2	1	1

2. **Đọc dữ liệu vào biến "dulieu"**

```
import pandas as pd
dulieu = pd.read_csv("housing_RT.csv", index_col=0)
dulieu.iloc[1:5,]
```

3. Sử **dụng nghi thức hold-out Phân chia tập dữ liệu huấn luyện**

```
from sklearn.model_selection import train_test_split
#X_train,X_test,y_train,y_test = train_test_split( dulieu.ix[:,1:5],dulieu.ix[:,0], test_size=1/3.0, random_state=100)
X_train,X_test,y_train,y_test = train_test_split( dulieu.iloc[:,1:5],dulieu.iloc[:,0], test_size=1/3.0, random_state=100)
X_train.iloc[1:5,]
X_test[1:5]
y_test[1:5]
```

```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(X_train, y_train)
```

4. **Dự báo và đánh giá mô hình**

- Dự đoán giá trị nhà

```
y_pred = regressor.predict(X_test)
y_test[1:5]
y_pred[1:5]
```

- Đánh giá kết quả dự đoán giá trị nhà thông qua chỉ số MSE và RMSE

```
from sklearn.metrics import mean_squared_error
err = mean_squared_error(y_test, y_pred)
err
import numpy as np
np.sqrt(err)
```