Lab: Tree, Bagging, RF

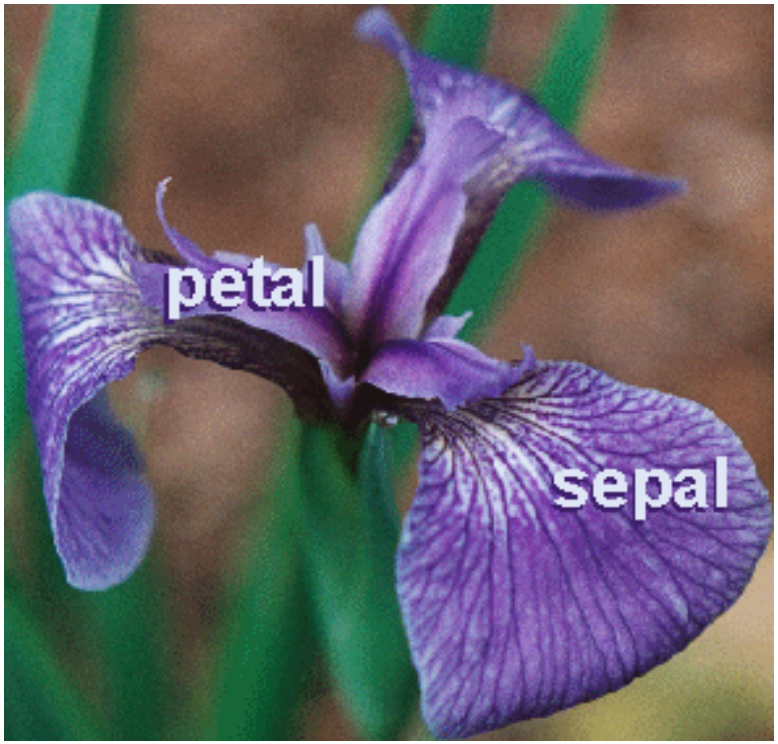
NỘI DUNG THỰC HÀNH

- Cây phân loại và hồi quy (CART)
- Các thuật toán học máy kết hợp (BAGGING & Random Forest)
- Các độ đo đánh giá hiệu quả của bộ phân lớp, mô hình hồi quy

Trees, RF: iris data, boston

Dữ liệu về hoa iris cung cấp đo lường liên quan đến chiều dài (sepal length, petal length), bề rộng (width)
– của 50 loại hoa
– từ 3 giống (setosa, versicolor, virginica)

- Mục tiêu: dùng đo lường để phân biệt các loài hoa



Dữ liệu phân tích: Boston

- **Boston data:** liên quan đến giá nhà đất
- Các biến số
  - **crim**: tỉ lệ tội phạm của thị trấn
  - **zn**: tỉ lệ khu đất có diện tich trên 25,000 feet vuông
  - **indus**: tỉ lệ doanh nghiệp tương đối lớn
  - **chas**: gần sông Charles (1=yes, 0=no)
  - **nos**: nồng độ nitric oxides (parts/10 triệu)
  - **rm**: số phòng trung bình mỗi nhà
  - **age**: tỉ lệ căn hộ (unit) xây trước 1940
  - **dis**: khoảng cách đến các trung tâm kĩ nghệ (tìm việc làm)

Dữ liệu phân tích: Boston

- **Boston data:** liên quan đến giá nhà đất
- Các biến số
  - **rad**: chỉ số gần xa lộ radial
  - **tax**: tỉ suất thuế tinh trên $10,000
  - **ptratio:** tỉ số học trò trên giáo viên của thị trấn
  - **black**: chỉ số về số người da đen trong thị trấn (Bk – 0.63)^2
  - **lstat**: tỉ lệ dân số thành phần kinh tế thấp
  - **PRICE:** trị giá nhà ($1000)

CART

- Yêu cầu: Mô tả về các bộ dữ liệu sử dụng trong các mô hình
- Đặt bài toán: Xây dựng cây phân loại và hồi quy cho thuộc tính nào của dữ liệu
- Ý nghĩa của cây hồi quy và cây phân loại

CART – Cây hồi quy

- Đọc dữ liệu: Boston

```
from sklearn.datasets import load_boston
boston = load_boston()
```

CART – Cây hồi quy

- Xây dựng cây hồi quy:

```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(max_depth=6)
DT_reg=regressor.fit(X_train, y_train)
```

- Lệnh này nhằm thực hiện công việc gì?

CART – Cây phân loại

- Lựa chọn tập dữ liệu: Iris

```
data = load_iris()
X = data.data
y = data.target
```

- Phân chia tập dữ liệu:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 50, test_size = 0.3)
```

CART – Cây phân loại

- Xây dựng cây phân loại bằng hàm tree:

```
from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier()

clf.fit(X_train,y_train)
```

CART – Cây phân loại

- Dựa vào kết quả nhận được, dự đoán việc phân loại cho tập test dùng hàm predict()

```
y_pred = clf.predict(X_test)
```

- Kiểm tra kết quả nhận được:

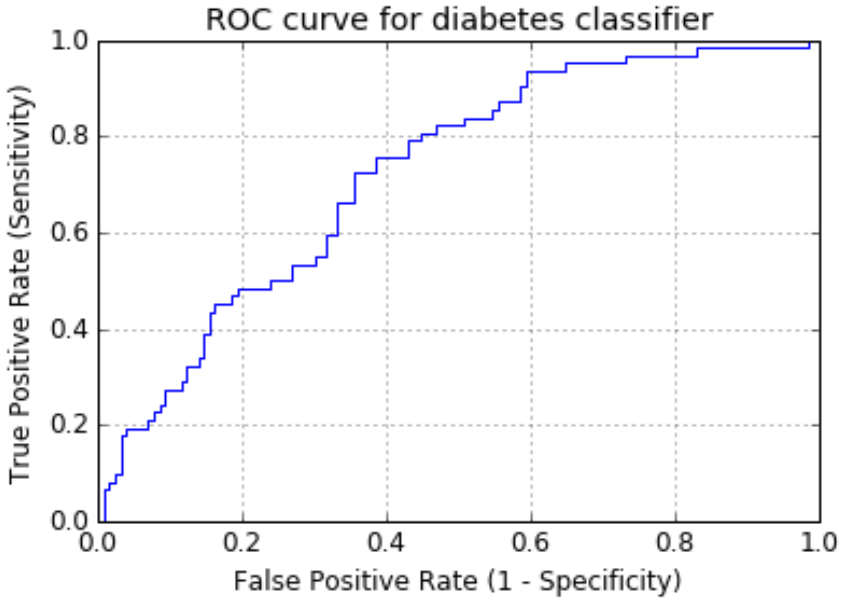
```
print(y_pred)
```

CART – Cây phân loại

- Các độ đo dùng đánh giá hiệu quả phân lớp
  - Conf. (Confusion) Matrix
  - Acc (Accuracy)
  - ROC (Receiver Operating Characteristic)
  - AUC (Area Under the Curve)

ROC

- Cho thấy sự ảnh hưởng của ngưỡng đến kết quả phân loại khi không cần thay đổi ngưỡng



AUC

- AUC là phần trăm của đồ thị ROC nằm dưới đường cong:
- Giá trị của AUC càng cao thì phân loại càng tốt