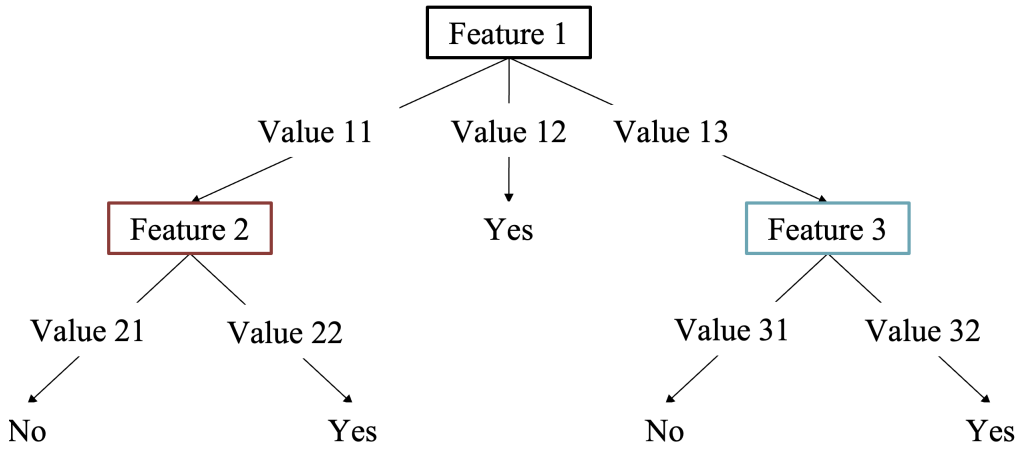
AI VIET NAM – COURSE 2024

Exercise: Decision Tree

Ngày 7 tháng 9 năm 2024

Decison Tree - Cây quyết định là phương pháp học có giám sát không tham số dựa vào ý tưởng xây dựng mô hình dạng cây để xấp xỉ dữ liệu huấn luyện. Cấu trúc cây quyết định được mô tả như hình sau:



Hình 1: Mô hình biểu diễn cây quyết định.

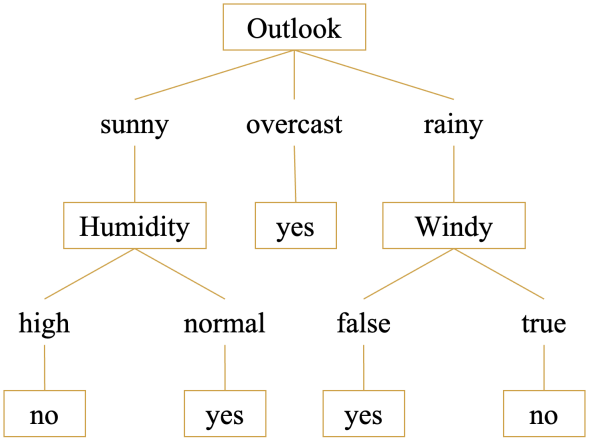
Cây quyết định có một số đặc trưng sau:

- Cây có các node là các thuộc tính của dữ liệu. Ví dụ như thuộc tính về Nhiệt độ.
- Các đường nối giữa các node là các giá trị của thuộc tính. Ví dụ thuộc tính nhiệt độ sẽ có các giá trị như: Cao, Trung Bình, Thấp.
- Các node lá đại diện cho các kết quả có thể có của mô hình. Ví dụ với bài toán phân loại nhị phân thì các node lá có thể nhận kết quả là '0' hoặc '1'.

Để xây dựng cây quyết định trên, chúng ta sử dụng thuật toán Iterative Dichotomiser (ID3) xây dựng theo chiều từ trên xuống dựa theo một số độ đo để xác định thuộc tính nào sẽ là node gốc và các thuộc tính ở node con là gì? Một số độ đo thường được sử dụng để đánh giá độ quan trọng của các thuộc tính như Gini Impurity hoặc Entropy / Information Gain cho bài toán classification và Variance / Sum of Squared Errors (SSE) cho bài toán regression.

Trong quá trình huấn luyện chúng ta sẽ đánh giá các độ đo và xem xét thuộc tính dựa vào các độ đo này có tốt để phân chia tập dữ liệu thành các tập con. Chúng ta sẽ tìm hiểu về các độ đo ở phần tiếp theo.

Ví dụ về cây quyết định sau khi được huấn luyện như sau:



Hình 2: Cây quyết định cho phân loại nhị phân.

1. Decision Tree for Classification

Độ đo Gini có thể được xác định bởi công thức:

$$Gini(D) = \frac{n_1}{n} Gini(D_1) + \frac{n_2}{n} Gini(D_2)$$

$$Gini(D_i) = 1 - \sum_{j=1}^{c} p_j^2$$

Độ đo Entropy được xác định bởi công thức:

$$Entropy(D) = \frac{n_1}{n} Entropy(D_1) + \frac{n_2}{n} Entropy(D_2)$$

$$Entropy(D_i) = -\sum_{j=1}^{c} p_j log_2 p_j$$

Trong đó D là tập dữ liệu ban đầu có n phần tử và có thể được phân chia thành 2 tập con là D_1 có n_1 phần tử và D_2 có n_2 phần tử. p_j là xác suất của các sample trong D_i thuộc vào class c.

Độ đo Information Gain được xác định bởi công thức:

$$Gain(D) = 1 - Entropy(D)$$

Để xác định được thuộc tính quan trọng với cây quyết định chúng ta có thể chọn *Entropy* hoặc *Gini* thấp nhất, *Gain* cao nhất.

Age	Likes English	Likes AI	Raise Salary
23	0	0	**0**
25	1	1	**0**
27	1	0	**1**
29	0	1	**1**
29	0	0	**0**

Hình 3: Bảng dữ liệu cho bài toán Classification.

Dựa vào công thức tính và bảng dữ liệu gồm các thuộc tính Age, Likes English, Likes AI và cột nhãn Raise Salary trả lời các câu hỏi sau đây:

Câu hỏi 1 Giả sử cho bài toán bộ dữ liệu D gồm các ví dụ được phân loại thành 2 lớp. Độ đo Entropy bằng 1 khi nào?

a) Số dữ liệu trong hai lớp bằng nhau

b) số dữ liệu trong lớp 1 bằng 3 lần số dữ liệu trong lớp 2

c) số dữ liệu trong lớp 1 bằng 2 lần số dữ liệu trong lớp 2

d) số dữ liệu trong lớp 1 bằng 4 lần số dữ liệu trong lớp 2

Câu hỏi 2 Tính giá trị Gini của các mẫu trong cột nhãn D = 'Raise Salary'?
a) Gini(D)=0.40
b) Gini(D)=0.44
c) Gini(D)=0.48
d) Gini(D)=0.46

Câu hỏi 3 Tính Gini của bộ dữ liệu khi thuộc tính 'Likes English' được chọn là node gốc?
a) Gini(Likes English)=0.40
b) Gini(Likes English)=0.43
c) Gini(Likes English)=0.45
d) Gini(Likes English)=0.47

Bảng dữ liệu gồm các thuộc tính Age, Likes English, Likes AI và cột nhãn Raise Salary. Trong đó sẽ có thuộc tính 'Age' thuộc kiểu dữ liệu số liên tục. Vì vậy với thuộc tính này, chúng ta sẽ sắp xếp lại các hàng theo thứ tự tăng dần theo cột 'Age'. Tương ứng với mỗi cặp giá trị liên tiếp chúng ta sẽ tính trung bình. Dựa vào các giá trị trung bình này để tính các độ đo. Ví dụ về phân chia các giá trị trong cột 'Age' mô tả như hình sau:

	Age	Likes English	Likes AI	Raise Salary
	23	0	0	**0**
24				
	25	1	1	**0**
26				
	27	1	0	**1**
28				
	29	0	1	**1**
29				
	29	0	0	**0**

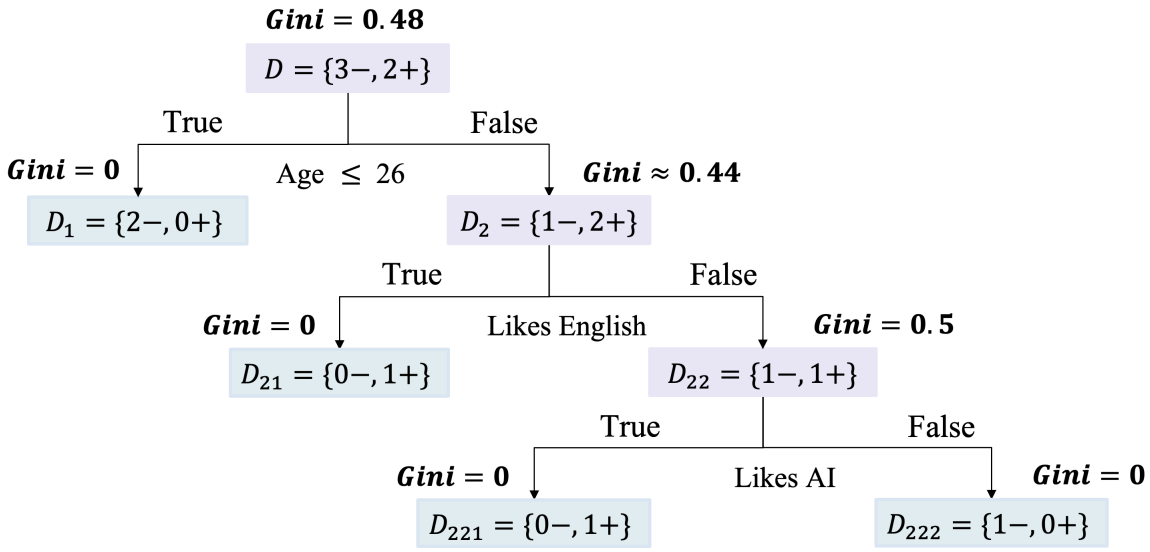
Hình 4: Phân loại trên cột dữ liệu liên tục 'Age'.

Câu hỏi 4 Tính Gini của bộ dữ liệu khi thuộc tính 'Age' được chọn là node gốc với điều kiện phân chia thành tập D_1 và D_2 là 'Age <= 26'?
a) Gini(Age <= 26)=0.25
b) Gini(Age <= 26)=0.26
c) Gini(Age <= 26)=0.27
d) Gini(Age <= 26)=0.28

Câu hỏi 5 Tính giá trị Entropy của các sample trong cột nhãn D = 'Raise Salary'?
a) Entropy(D)=0.99
b) Entropy(D)=0.97
c) Entropy(D)=0.95
d) Entropy(D)=0.93

Câu hỏi 6 Tính Gain của bộ dữ liệu khi thuộc tính 'Likes English' được chọn là node gốc?
a) Gain(Likes English)=0.048
b) Gain(Likes English)=0.038
c) Gain(Likes English)=0.028
d) Gain(Likes English)=0.018

Ví dụ về cây quyết định sau khi được huấn luyện như sau:



Hình 5: Cây quyết định cho bộ dữ liệu trên.

Tiếp theo chúng ta sẽ vào phần code sử dụng thư viện 'sklearn' và phân loại trên bộ dữ liệu IRIS.

Câu hỏi 7 Dòng lệnh nào sau đây để tải về bộ dữ liệu Iris từ thư viện 'sklearn' là?
a) iris_X, iris_y = datasets.load_iris(return_X_y=True)
b) iris_X, iris_y = datasets.load_iris()
c) iris_X, iris_y = datasets.load_iris_data(return_X_y=True)
d) iris_X, iris_y = datasets.load_iris_data()

Câu hỏi 8 Sắp xếp các đoạn code phân loại trên bộ dữ liệu Iris dựa vào Decision Tree:

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier

Paragraph A:
# Train
dt_classifier.fit(X_train, y_train)

Paragraph B:
# Define model
dt_classifier = DecisionTreeClassifier()

Paragraph C:
# Load the diabetes dataset
```

```
iris_X, iris_y = # From question 7
# Split train:test = 8:2
X_train, X_test, y_train, y_test = train_test_split(
    iris_X, iris_y,
    test_size=0.2,
    random_state=42)

Paragraph D:
# Preidct and evaluate
y_pred = dt_classifier.predict(X_test)
accuracy_score(y_test, y_pred)
```

Thứ tự đúng là
a) C - B - A - D
b) A - B - D - C
c) D - C - A - B
d) A - C - D - B

2. Decision Tree for Regression

Với bài toán Regression. Để chọn được thuộc tính tốt phân chia bộ dữ liệu và xây dựng cây, chúng ta sẽ sử dụng độ đo SSE. Độ đo Sum of Squared Error (SSE) được xác định bởi công thức:

$$SSE(D) = SSE(D_1) + SSE(D_2)$$

$$SSE(D_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} (x_j - \overline{x}_i)^2$$

Trong đó D là tập dữ liệu ban đầu có n phần tử và có thể đuợc phân chia thành 2 tập con là D_1 có n_1 phần tử và D_2 có n_2 phần tử. $\overline{x}_i$ là giá trị trung bình của các phần tử trong tập D_i.

Để xác định được thuộc tính quan trọng với cây quyết định chúng ta chọn SSE với giá trị cao nhất.

Age	Likes English	Likes AI	Salary
23	0	0	**200**
25	1	1	**400**
27	1	0	**300**
29	0	1	**500**
29	0	0	**400**

Hình 6: Bảng dữ liệu cho bài toán Regression.

Dựa vào công thức tính và bảng dữ liệu gồm các thuộc tính Age, Likes English, Likes AI và cột giá trị thực tế Salary trả lời các câu hỏi sau đây:

Câu hỏi 9 Tính SSE của bộ dữ liệu khi thuộc tính 'Likes AI' được chọn là node gốc?
a) SSE(Likes AI)=9167
b) SSE(Likes AI)=2500
c) SSE(Likes AI)=6667
d) SSE(Likes AI)=5000

Câu hỏi 10 Tính SSE của bộ dữ liệu khi thuộc tính 'Age' được chọn là node gốc với điều kiện phân chia thành tập D_1 và D_2 là 'Age <= 24'?
a) SSE(Age <= 24)=3000
b) SSE(Age <= 24)=4000
c) SSE(Age <= 24)=5000
d) SSE(Age <= 24)=6000

Câu hỏi 11 Sắp xếp các đoạn code phân loại trên bộ dữ liệu CPU Machine dựa vào Decision Tree Regressor:

```
from sklearn.datasets import fetch_openml
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.tree import DecisionTreeRegressor

Paragraph A:
# Train
tree_reg.fit(X_train, y_train)

Paragraph B:
# Define model
tree_reg = DecisionTreeRegressor()

Paragraph C:
# Load dataset
machine_cpu = fetch_openml(name='machine_cpu')
machine_data = machine_cpu.data
machine_labels = machine_cpu.target
# Split train:test = 8:2
X_train, X_test, y_train, y_test = train_test_split(
    machine_data, machine_labels,
    test_size=0.2,
    random_state=42)

Paragraph D:
# Preidct and evaluate
y_pred = tree_reg.predict(X_test)
mean_squared_error(y_test, y_pred)
```

Thứ tự đúng là
a) C - B - A - D
b) A - B - D - C
c) D - C - A - B
d) A - C - D - B

- Hết -