

# 7주차 정리 노트

202184026 서동현

201904025 김성욱

## CH.06

### 결정 트리

- ➔ 분류, 회귀 작업, 다중 출력 작업이 가능한 머신러닝 알고리즘
- ➔ 매우 복잡한 데이터셋도 학습 가능(장점)
- ➔ 랜덤 포레스트의 기본 구성 요소
- ➔ 데이터 전처리 불필요(장점)

### 결정 트리 학습과 시각화

- ➔ 트리 구성
  - Node : 가지치기가 시작되는 지점
  - Root node : 맨 상단에 위치한 노드
  - Leaf node : 더 이상의 가지치기가 발생하지 않는 노드

**max\_depth**: 결정 트리 최대 깊이 지정

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris()
X = iris.data[:, 2:] # 꽃잎 길이와 너비
y = iris.target

tree_clf = DecisionTreeClassifier(max_depth=2, random_state=42)
tree_clf.fit(X, y)
```

- ➔ **max\_depth**가 2이기 때문에 연속 가지치기 최대 2번까지 가능

### 결정 트리 학습결과 시각화

```
from graphviz import Source
from sklearn.tree import export_graphviz

export_graphviz(
    tree_clf,
    out_file=os.path.join(IMAGES_PATH, "iris_tree.dot"),
    feature_names=iris.feature_names[2:],
    class_names=iris.target_names,
    rounded=True,
    filled=True
)

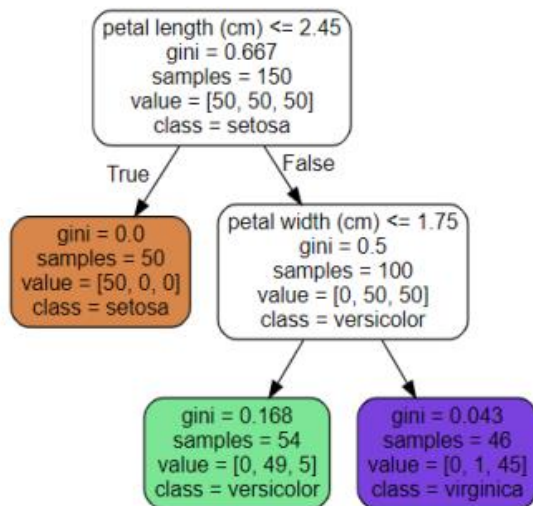
Source.from_file(os.path.join(IMAGES_PATH, "iris_tree.dot"))
```

**Gini** : 해당 노드의 불순도 측정값(0 ~ 1범위), 비용함수에 사용

**Samples** : 해당 노드 결정에 사용된 샘플 수

**Values** : 해당 마디 결정에 사용된 샘플을 클래스 별로 구분한 결과

**Class** : 각 클래스별 비율을 계산하여 가장 높은 비율에 해당하는 클래스 선정



▲ 그림 6-1 붓꽃 결정 트리

예측하기

클래스 확률 추정 : 결정 트리는 한 샘플이 특정 클래스 k에 속할 확률을 추정 할 수 있다.

## 클래스 확률 추정 코드

```
tree_clf.predict_proba([[5, 1.5]])
array([[0.          , 0.90740741, 0.09259259]])

tree_clf.predict([[5, 1.5]])
array([1])
```

CART 훈련 알고리즘

- ➔ 탐욕적 알고리즘 : 여러 경우 중 하나를 결정해야 할 때마다 그 순간에 최적이라고 생각 되는 것을 선택해 나가는 방식.
- ➔ 비용함수를 최소화하는 특성 k와 해당 특성의 임계값 tk를 결정

계산 복잡도

지니 불순도 vs 엔트로피("entropy")

엔트로피 : 분자의 무질서함을 측정하는 열역학의 개념

- ➔ 분자가 안정되고 질서 정연하면 엔트로피가 0에 가까움
- ➔ 머신러닝 불순도 측정 방법: 어떤 세트가 한 클래스의 샘플만 담고 있다면 엔트로피 0

규제 매개변수

- ➔ 비매개변수:
  - 훈련되기 전에 파라미터 수가 결정되지 않는 모델
  - 모델 구조가 데이터에 맞춰져서 고정되지 않고 자유로움
  - 과대적합 위험 높음
- ➔ 매개변수:
  - 미리 정의된 모델 파라미터
  - 자유도 제한
  - 과대적합 위험 줄어듦, 과소적합 위험 커짐

사이킷런 DecisionTreeClassifier 규제 매개변수

- max\_depth: 결정 트리의 최대 높이 제한
- min\_samples\_split: 노드를 분할하기 위해 필요한 최소 샘플 수
- min\_samples\_leaf: 리프 노드가 가지고 있어야 할 최소 샘플 수
- min\_weight\_fraction\_leaf:
  - 샘플 별로 가중치가 설정된 경우: 가중치의 전체 합에서 해당 리프 노드에 포함된 샘플의 가중치의 합이 차지하는 비율
  - 샘플 별로 가중치가 없는 경우: min\_samples\_leaf와 동일한 역할 수행
- max\_leaf\_nodes: 허용된 리프 노드의 최대 개수
- max\_features: 각 노드에서 분할 평가에 사용될 수 있는 최대 특성 수
- 규제를 높이는 방법
  - min\_ 접두사 사용 규제: 매개변수를 증가시킴.
  - max\_ 접두사 사용 규제: 매개변수를 감소시킴.

```
from sklearn.datasets import make_moons
Xm, ym = make_moons(n_samples=100, noise=0.25, random_state=53)

deep_tree_clf1 = DecisionTreeClassifier(random_state=42)
deep_tree_clf2 = DecisionTreeClassifier(min_samples_leaf=4, random_state=42)
deep_tree_clf1.fit(Xm, ym)
deep_tree_clf2.fit(Xm, ym)
```

회귀(결정트리)

결정 트리 회귀 모델: 사이킷런 DecisionTreeRegressor를 사용

```
from sklearn.tree import DecisionTreeRegressor

tree_reg = DecisionTreeRegressor(max_depth=2, random_state=42)
tree_reg.fit(X, y)
```

Samples : 해당 마디에 속한 훈련 샘플 수

Value : 해당 마디에 속한 훈련 샘플의 타깃값

Mse : 해당 마디에 속한 훈련 샘플의 평균제곱오차

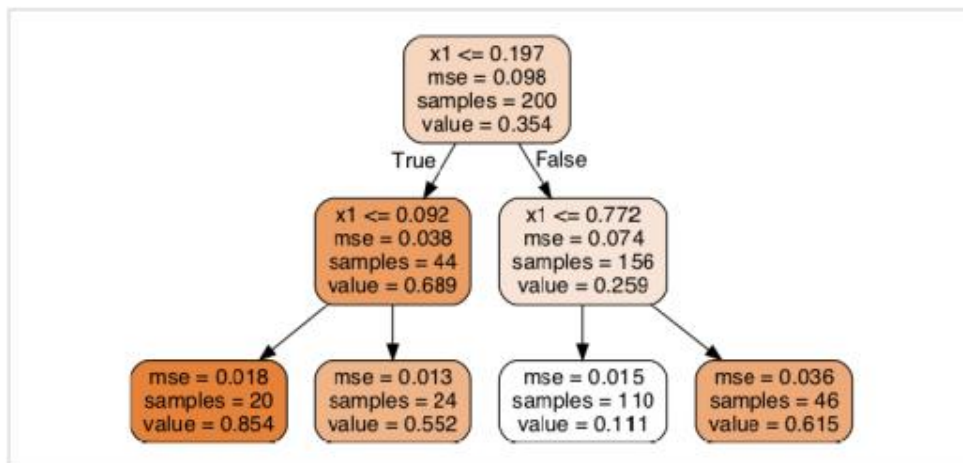


그림 6-4 회귀 결정 트리

결정 트리 불안정성

: 결정 트리는 여러 용도로 사용 가능하고, 성능이 매우 우수하지만 훈련 세트에 민감하게 반응함

1. 훈련 세트의 회전에 민감
2. 훈련 세트의 작은 변화에 매우 민감하다.

## <연습문제>

김성욱

### 연습문제

+ 코드

```
[2] 1 from sklearn.datasets import make_moons
    2
    3 # X, y를 n_samples(표본 데이터)를 1000개로 하고 잡음은 0.4로 설정한다
    4 X, y = make_moons(n_samples=1000, noise=0.4)

[10] 1 from sklearn.model_selection import train_test_split
    2 X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
    3 # Train, Test = 8 : 2 --> test_size = 0.2
```

### ▸ train\_test\_split에서 random\_state를 사용한 이유

1. 호출할 때마다 동일한 학습/테스트용 데이터 세트를 생성하기 위해 주어지는 난수 값.
2. train\_test\_split는 랜덤으로 데이터를 분리하므로 train\_test\_split를 설정하지 않으면 수행할 때마다 다른 학습/테스트 데이터 세트가 생성된다. 따라서 random\_state를 설정하여 수행 시 결과값을 동일하게 맞춰준다.
3. 이때 random\_state에는 어떤 숫자를 적든 그 기능은 같기 때문에 어떤 숫자를 적든 상관없다.

```
1 from sklearn.tree import DecisionTreeClassifier
2 from sklearn.model_selection import GridSearchCV
3 from sklearn.metrics import accuracy_score
4
5 params = {'max_leaf_nodes': list(range(2,5)),
6           'min_samples_split': list(range(2,3))}
7
8 gs = GridSearchCV(DecisionTreeClassifier(random_state=42),params,cv=3)
9 gs.fit(X_train,y_train)
10 em = gs.best_estimator_
11 pred = em.predict(X_test)
12 accuracy_score(y_test,pred)
13
```

0.87

## DecisionTreeClassifier + GridSearchCV

params(파라미터 설정) - 중요!!

- max\_leaf\_nodes :  
range(2,3) - accuracy: 0.795 ->  
range(2,4) - accuracy: 0.845 ->  
range(2,5) - accuracy: 0.87 -> 5이상 부터는 결과가 바뀌지 않았다.
- min\_samples\_split  
range의 크기에 따라서 accuracy가 변동하지 않았다. GridSearchCV
- DecisionTreeClassifier(random\_state) : 랜덤 시드를 사용해 결과값을 동일하게 맞춰줌
- cv(int) : 교차 검증을 위해 분할되는 학습/테스트 세트의 개수 지정

모델 학습 및 예측

```
gs.fit(X_train,y_train)
em = gs.best_estimator_
pred = em.predict(X_test)
accuracy_score(y_test,pred)
```

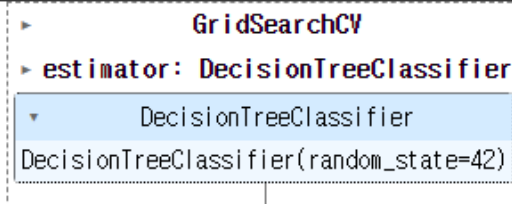
## <연습문제>

서동현

```
1 from sklearn.datasets import make_moons
2 X, y = make_moons(n_samples=1000, noise=0.4)
3
```

```
1 from sklearn.model_selection import train_test_split
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
3
```

```
1 # 튜닝할 하이퍼파라미터 그리드 정의
2 param_grid = {
3     'max_leaf_nodes': [None, 2, 5, 10, 20, 30]
4 }
5
6 # GridSearchCV를 사용하여 모델 튜닝
7 dt_classifier = DecisionTreeClassifier(random_state=42)
8 grid_search = GridSearchCV(dt_classifier, param_grid, cv=5)
9 grid_search.fit(X_train, y_train)
10
11
12
```



```
GridSearchCV
estimator: DecisionTreeClassifier
DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

GridSearchCV만 사용한 이유?

'max\_leaf\_nodes'를 중심으로 모델의 성능을 향상시키기 위해 단순하고 직관적인 방법을 선택

5-겹 교차 검증 (cv=5)을 적용하여 최적의 하이퍼파라미터를 찾습니다

모델의 일반화 성능을 평가하고 최적의 하이퍼파라미터를 결정합니다.

```
1 # 찾은 매개변수를 사용해 전체 훈련 세트에 대해 모델을 훈련
2 best_classifier = grid_search.best_estimator_
3
4
5 # 최적의 모델로 테스트 데이터에 대한 성능 측정
6 test_accuracy = best_classifier.score(X_test, y_test)
7 print("Test Accuracy:", test_accuracy)
```

```
Test Accuracy: 0.855
```

## 질문과 답변

1. CART알고리즘의 시간 복잡도는? (서동현)

↳ 정해져있는 것은 아닌 것 같다 (김성욱)

2. 지니 불순도와 엔트로피 중 어떤 지표를 사용하는 것이 좋은가 (서동현)

↳ 지니 불순도는 카테고리에 사용되고, 엔트로피는 복잡한 정보?

---

1. Decision Tree를 사용하는 이유? (김성욱)

↳ 직관적이며 해석하기 쉽다 (서동현)

2. 결정 트리는 왜 데이터 전처리가 불필요 한지? (김성욱)

↳ 뭔가 민감하지 않은 것이라서? (서동현)