# 머신러닝 6주차 요약: 결정 트리 (Decision Tree)

## 1. 의사결정트리(Decision Tree)란?

데이터를 나무처럼 분기하여 분류 또는 예측하는 지도학습 모델.  
- 분류: 가장 많은 클래스를 선택.  
- 회귀: 평균값을 예측.  
구조:  
- Root Node: 나무의 시작.  
- Parent Node: 상위 노드.  
- Child Node: 하위 노드.  
- Leaf Node: 최종 결과 노드.  
- Depth: 나무의 깊이.

## 2. 결정트리 훈련 과정

1단계: 데이터를 반복적으로 분할해 트리를 생성.  
2단계: 필요 없는 가지를 잘라내어 과적합 방지.  
3단계: 최종 트리로 예측 수행.

## 3. CART 알고리즘

Classification And Regression Tree 알고리즘.  
분할 시 비용 함수를 최소화하는 방향으로 데이터를 나눔.  
목표: 두 그룹이 최대한 순수하게 나눠지도록 함.

## 4. 지니 불순도 vs. 엔트로피

- 지니 불순도: 무작위로 샘플을 뽑았을 때 다른 클래스를 가질 확률.  
- 엔트로피: 데이터의 혼잡도를 수치화한 것.  
둘 다 그룹을 깨끗하게 나누기 위한 척도입니다.

## 5. 결정트리 규제 (Overfitting 방지)

트리는 과적합되기 쉽다.  
규제 방법:  
- max\_depth: 트리의 최대 깊이 제한.  
- min\_samples\_split: 분할하기 위한 최소 샘플 수 지정.  
- min\_samples\_leaf: 리프 노드에 필요한 최소 샘플 수 설정.

## 6. 회귀용 결정트리 (DecisionTreeRegressor)

연속적인 값을 예측할 때 사용.  
리프노드에 있는 샘플들의 평균값을 반환.  
회귀 트리도 과적합 위험이 있으므로 규제가 필요함.

## 7. 결정트리의 단점

- 훈련 데이터에 매우 민감.  
- 높은 분산(Variance) 문제 발생 가능.  
→ 랜덤 포레스트(Random Forest)로 여러 트리를 조합하여 해결.

## 요약

- 결정트리는 해석이 쉽고 빠르게 예측할 수 있다.  
- 하지만 규제를 하지 않으면 과적합에 취약하다.  
- 분류와 회귀 모두 사용할 수 있음.  
- 랜덤 포레스트를 통해 약점을 보완할 수 있다.