기초 이론부터 실무 실습까지

머신 러닝 익히기

programmers

Part 05. SVM과 Decision Tree

정 정 민

programmers

Chapter 13. Decision Tree

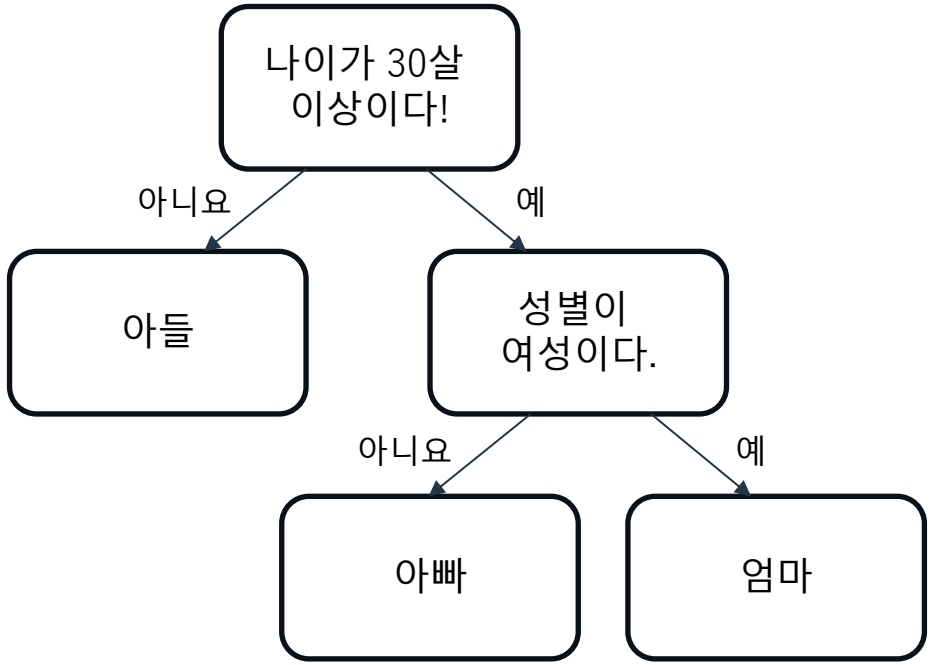
1. Tree 용어
2. 분류 문제를 위한 Decision Tree
3. 회귀 문제를 위한 Decision Tree
4. 주의 사항

Tree 용어

3인 가족 나누기!

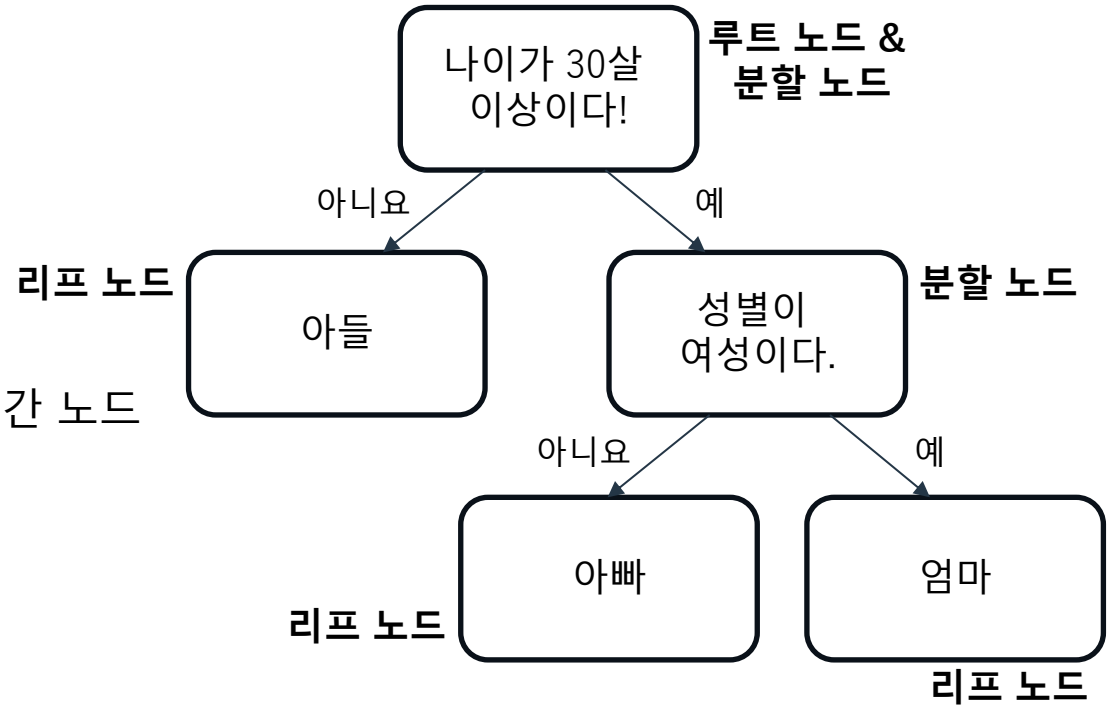
- 어떤 기준을 사용해 가족 구성원을 나눠볼까요?

- 구성원을 나누기 위한 조건은 다양함
- 어떤 조건을 선택하는지에 따라 빠르게 나눌 수 있고, 더디게 나눌 수 있음



Tree 구조의 용어

- **노드**
  - 데이터에 대한 특정 질문이나 조건
  - 데이터를 분류하는 과정에서 사용
- **엣지**
  - 노드와 노드를 연결하는 선
  - 상위 노드의 특정 질문에 대한 가능한 답변
- **루트 노드**
  - 트리의 가장 상단에 위치한 노드
  - 분류 또는 예측을 시작하는 지점
- **분할 노드** (= **결정 노드**)
  - 데이터를 더 작은 하위 집합으로 나누는 데 사용되는 중간 노드
- **리프 노드** (= **터미널 노드**)
  - 트리에 말단에 위치한 노드
  - 더 이상의 분기가 없고 자식 노드를 갖지 않음



분류 문제를 위한 Decision Tree

강의 실습 코드 : 링크

결정 기준 (Decision Criteria)

- 데이터를 분할하는 기준을 결정하는데 사용되는 방법론

- 트리의 각 단계에서 최적의 분할을 찾기 위해 사용되며
- 이를 통해 트리의 깊이와 복잡성을 관리할 수 있음

- 좋은 결정 기준은 트리를 더욱 간결하고 효율적으로 만들며
- 과적합을 방지하고 일반화 성능을 향상시킴!

- **분류 과정**에서 사용되는 결정 기준
  - 정보 이득
  - 지니 불순도

- **회귀 과정**에서 사용되는 결정 기준
  - MSE 최소화

엔트로피 (Entropy)

- 어떤 상황이나 현상이 품고 있는 불확실성을 의미하며, 포함하는 정보의 양과 반비례
  - 엔트로피가 크다 → 불확실성이 크다 → 정보량이 적다
  - 반대로, 엔트로피가 작으면 → 불확실성이 작고, 알고 있는 정보가 많다!

- 예를 들어, 두 사람이 가위 바위 보를 하는 상황에서
- **[상황 A] 두 사람이 뭘 낼지 모를 때**
  - 어떠한 정보도 없음
  - 따라서 불확실성이 큼
  - 결국 엔트로피가 크다
- **[상황 B] 한 사람이 뭘 낼지 알고, 한 명은 뭘 낼지 모를 때**
  - 한 명의 정보는 알고 있음
  - 불확실성이 있지만, 상황 A보다는 아님
  - 엔트로피가 상황 A보다는 작음!

엔트로피 (Entropy) 수식

- 엔트로피는 수치적으로 구할 수 있으며 아래와 같음

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

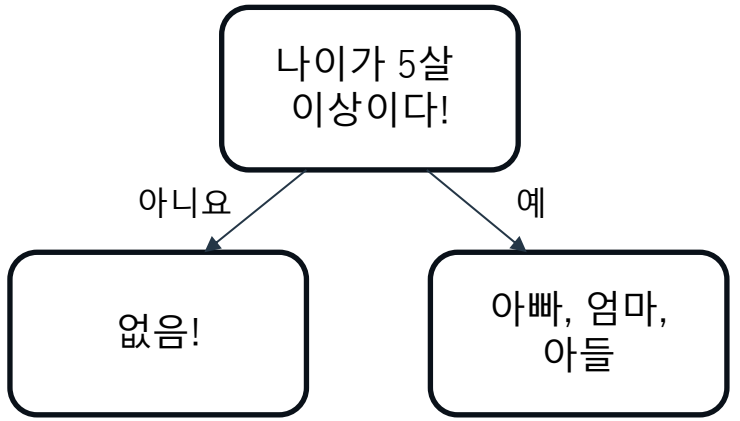
- $P(x_i)$: 각 사건의 발생 확률, 불확실성이 높은 상황일수록, 다양한 사건의 발생 확률이 균일!
- $\log_2 P(x_i)$: 이진 로그를 사용해 '비트(bit)' 단위의 정보량을 표현
- $-\sum$: 발생할 각 사건을 모두 더한 뒤, $\log_2$ 값을 양수로 만들기 위해 $-$를 취함

- 앞의 가위 바위 보 예에서
  - 상황 A : 3.170 (p = 1/9)
  - 상황 B : 1.585 (p=1/3)
    - 한 사람의 엔트로피는 0

정보 이득 (Information Gain)

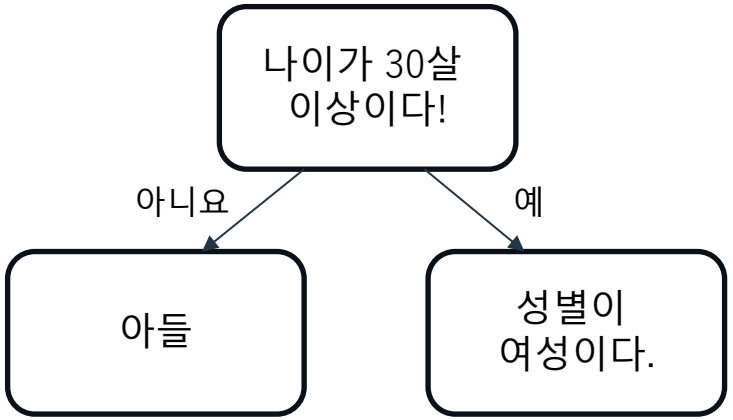
- 각 상황은 특정한 정도의 엔트로피를 갖고 있음을 알 수 있었음

- 따라서 각 노드에 포함되는 데이터의 순도에 따라 엔트로피가 계산될 수 있음
  - 노드 안에 서로 다른 클래스의 데이터가 많이 섞여 있으면 순도 ↓
  - 같은 클래스의 데이터가 모여 있다면 순도 ↑
  - '아빠' '엄마' '아들' 을 갖는 노드는 '아들'만 갖는 노드보다 순도가 낮음
    - 불확실성이 크고, 정보가 적으며, 엔트로피가 크다!



- 정보 이득 (information gain)이란,
  - 부모 노드와 자식 노드들의 엔트로피를 계산해
  - 엔트로피가 낮아지는 방향으로 결정 경계를 선정하는 것을 의미함
    - 즉, 정보 이득을 최대화 하는 방향으로!

$$InfoGain = Entropy_{Parent} - \sum \frac{N_{child}}{N_{total}} Entropy_{Child}$$

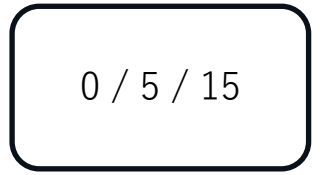


엔트로피 계산 예시

- A ~ C 각 노드의 엔트로피 계산
  - A : $-\left(\frac{0}{20}\log_2\frac{0}{20}+\frac{5}{20}\log_2\frac{5}{20}+\frac{15}{20}\log_2\frac{15}{20}\right)=0.8113$
  - B : $-\left(\frac{7}{20}\log_2\frac{7}{20}+\frac{6}{20}\log_2\frac{6}{20}+\frac{7}{20}\log_2\frac{7}{20}\right)=1.5813$
  - C : $-\left(\frac{20}{20}\log_2\frac{20}{20}+\frac{0}{20}\log_2\frac{0}{20}+\frac{0}{20}\log_2\frac{0}{20}\right)=0$

- 각 노드의 클래스 분포를 기준으로 엔트로피를 계산할 수 있음
- 엔트로피가 클 수록 데이터가 고르게 분포 (순도가 낮음)

- 엔트로피가 작아지는 방향으로 노드를 만드는 결정 경계를 생성해야 함!



A 노드

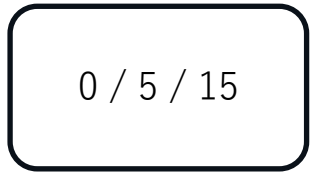
B 노드

C 노드

지니 불순도 (Gini Impurity)

- 데이터 집합의 순도를 측정하는 또 다른 방법
- 데이터 안에 존재하는 클래스 분포의 불균형을 평가하는 방법

$$Gini(S) = 1 - \sum_{i=1}^{n} p_i^2$$



A 노드

- p_i : 데이터 집합 안에 존재하는 i 번째 클래스가 나타나는 확률
- 0 이상 1 미만의 값을 갖으며,
  - 0 : 모든 데이터가 하나의 클래스에 속함. 제일 순도가 높은 상태
  - 1에 가까운 값 : 모든 클래스의 데이터가 고루 섞인 상태로 불순도가 제일 높음

B 노드

- A ~ C 각 노드의 지니 불순도
  - A : $1 - \left(\frac{0}{20}^2 + \frac{5}{20}^2 + \frac{15}{20}^2\right) = 0.375$
  - B : $1 - \left(\frac{7}{20}^2 + \frac{6}{20}^2 + \frac{7}{20}^2\right) = 0.665$
  - C : $1 - \left(\frac{20}{20}^2 + \frac{0}{20}^2 + \frac{0}{20}^2\right) = 0$

C 노드

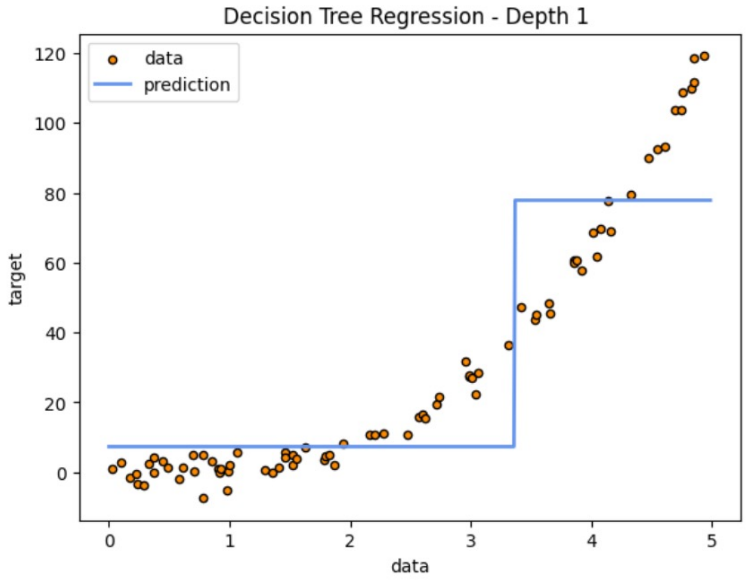
회귀 문제를 위한 Decision Tree

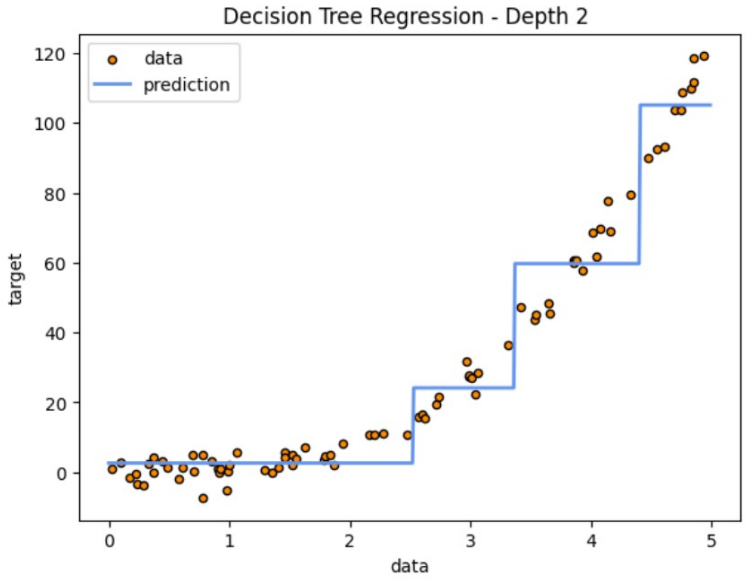
강의 실습 코드 : 링크

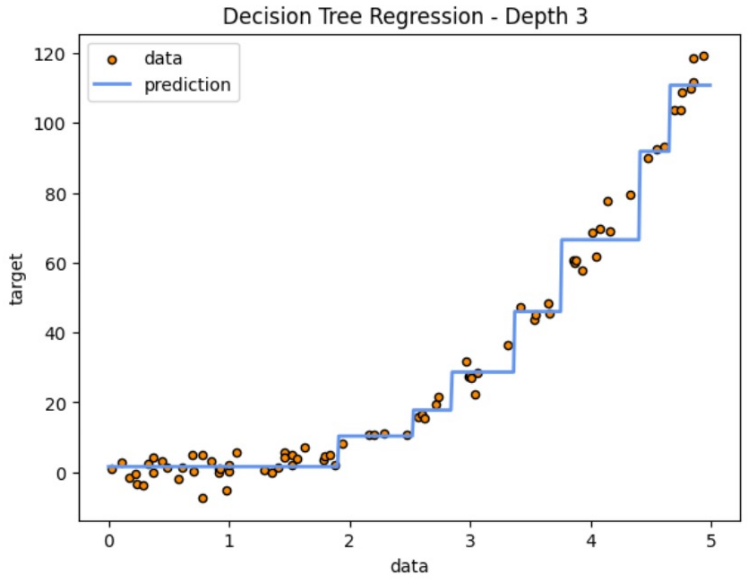
MSE 최소화 방식

- Decision Tree에서 회귀 문제를 푸는 주요한 방식 중 하나

- 각 노드에서 실제 정답과 예측 값 사이의 평균 제곱 오차(MSE)의 평균을 계산하고
- 이 값을 최소화하는 노드를 찾아가는 방식으로 Tree가 만들어짐





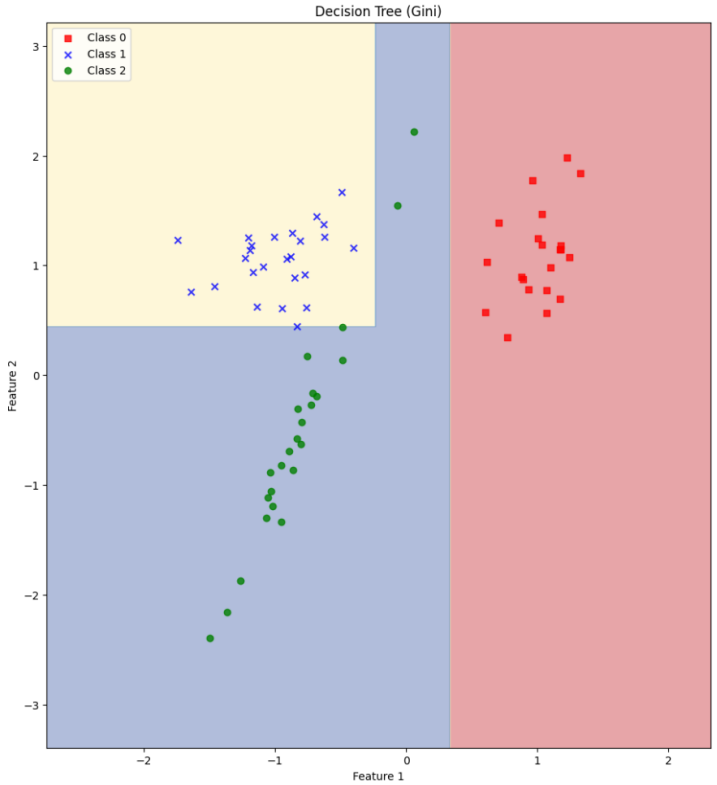


주의사항

Tree 사용 과정에서의 주의사항

- 트리 구조는 상당히 많은 장점을 갖고 있음
  - 해석 용이성 & 사용 편리함
  - 적당히 괜찮은 성능
  - 스케일링에 둔감함 등

- 하지만 트리 기반 모델은
  - 축에 수직인 방향으로 데이터가 분할됨
    - 따라서 축에 수직한 데이터는 쉽게 해결하지만
      경계면이 회전이 되어있다면 구불구불한 경계면이 생성됨
    - 일반화에 어려움이 있을 수 있음
    - 필요시 주성분 분석을 통한 데이터 회전이 필요할 수 있음
  - 데이터 노이즈에 굉장히 민감하며
    - 특정 데이터의 추가가 전체 모델 결과에 큰 변화를 줄 수 있음
  - Depth가 깊다면 강한 Overfit의 위험이 큼



E.O.D