Decision Tree

의사결정규칙을 나무구조로 나타내어 전체 데이터를 소집단으로 분류하거나 예측하는 방법

데이터 사이에 존재하는 패턴을 예측 가능한 규칙들의 조합으로 나타내는데, 형태가 나무와 유사

초기의 지점을 root node, 끝마디를 terminal node, 중간마디를 intermediate node라고 함

통과하는 노드가 늘어날수록, 조건에 부합하는 데이터의 수는 줄어들게 됨

terminal node의 데이터들의 합 = terminal node 데이터

좋은 Decision Tree 모델이란 같은 정확도이면서, 일반화를 조금 더 잘하는 심플한 모델을 의미

- 즉, 최대한 한가지 클래스만 가지도록 분류하는 모델이 좋음

좋은 노드를 분류할 수 있는 기준은 불순도를 통해서 정의

불순도를 측정하는 지표에는 엔트로피, 지니지수가 있음

ID3 알고리즘

- 무질서도, 즉 데이터의 불확실성을 수치화하는 엔트로피 지수를 이용한 알고리즘
- 상위 노드의 엔트로피에서 하위 노드의 엔트로피를 뺀 값인 Information Gain이 크게 나오는 변수를 기준으로 선택
- Information Gain 값이 클수록, 엔트로피를 많이 줄였다는 의미로 불확실성이 최소가 됨을 의미

CART 알고리즘

- 데이터의 통계적 분산정도를 정량화해서 표현한 값인 지니지수를 이용한 알고리즘
- Binary split을 전제로 분석, 데이터의 대상 속성을 얼마나 잘못 분류할지를 계산
- 지니지수가 감소할수록, 불순도가 감소하기 때문에 지니지수를 감소시키는 방향으로 분류

feature가 연속형일 경우에는 각 feature에 대해 오름차순으로 정렬하고, class가 변하는 지점을 찾은 뒤에 경계의 평균값을 기준으로 분할하여 지니지수와 엔트로피를 계산하는 방법을 이용

가지치기

- 모든 terminal node 의 순도가 100%일 경우에는 과적화 위험이 발생해 일반화능력이 떨어짐
- 과적합을 방지하기 위하여 적절한 수준에서 terminal node 를 결합해주는 방법
- 트리의 최대 깊이나 분기점의 최소 개수를 미리 지정하는 사전 가지치기
- 트리를 만든 뒤에 데이터 포인트가 적은 노드를 삭제/병합하는 사후 가지치기

의사결정나무의 장단점

장점

- 결과를 해석하고, 이해하기 쉬움
- 비모수적 모형이기 때문에 선형성, 정규성, 등분산성 등의 가정이 필요하지 않음
- 데이터 가공할 필요가 거의 없음

단점

- 연속형 변수를 비연속적 값으로 취급하기 때문에 분리의 경계점 부근에서 예측 오류가 클 가능성이 있음
- 데이터의 특성이 특정 변수에 수직/수평적으로 구분되지 못할 때, 성능이 떨어지고 트리가 복잡해짐
- 과적합 위험이 높음
- 중간 단계에서 오류 발생시, 다음 단계로 에러가 계속 전파
- 적은 개수의 노이즈에도 크게 영향을 받음

이러한 단점을 해결하기 위해 앙상블 방법을 이용