CHAPTER6. 결정 트리

- 결정 트리 학습
- 예측하기
- 클래스 확률 추정
- CART 훈련 알고리즘
- 계산 복잡도
- 지니 불순도 또는 엔트로피
- 규제 매개변수
- 회귀
- 불안정성

6.1 결정 트리 (Decision Tree) 학습

- 결정 트리는 분류와 회귀 그리고 다중 출력 작업도 가능한 다재다능한 머신러닝 알고리즘입니다.
- 복잡한 데이터 셋도 학습할 수 있고 랜덤 포레스트의 기본 구성 요소입니다.

6.2 예측하기

- 루트 노드 : 자식 노드를 가지고 있는 노드
- 리프 노드 : 자식 노드가 없는 노드
- value : 노드에서 각 클래스에 얼마나 많은 훈련 샘플이 있는지 알려줍니다.
- gini : 각 노드의 불순도(impurity)를 측정합니다. 한 노드의 모든 샘플이 같은 클래스에 속해있으면 gini가 0입니다.

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2$$

- $P_{i,k}$는 i 번째 노드에 있는 훈련 샘플 중 클래스 K에 속한 샘플의 비율입니다.

6.3 클래스 확률 추정

- 결정 트리는 한 샘플이 특정 클래스 K에 속할 확률을 추정할 수도 있습니다. (.predict_proba)
- 먼저 이 샘플에 대해 리프 노드를 찾기 위해 트리를 탐색합니다.
- 그리고 그 노드에 있는 클래스 K의 훈련 샘플의 비율을 반환합니다.

6.4 CART 훈련 알고리즘

- 결정 트리를 훈련시키기 위해 CART (Classification And Regression Tree) 알고리즘을 사용합니다.
- 먼저 훈련 세트를 한의 특성 k의 임곗값 t_k를 사용해 두 개의 서브셋으로 나눕니다.
- 훈련 세트를 성공적으로 둘로 나눴으면 같은 방식으로 서브셋을 또나누고 그 다음엔 서븝셋의 서브셋을 나누는 식으로 반복합니다.
- k와 t_k를 고르는 방법은 크기가 드른 가중치가 적용된 가장 순수한 서브셋으로 나눌 수 있는 (k, t_k) 짝을 찾습니다.

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$$

- 위 식은 알고리즘이 최소화해야 하는 비용 함수입니다.
- $G_{left/right}$ 는 왼쪽 / 오른쪽 서브셋의 불순도입니다.
- $m_{left/right}$ 는 왼쪽 / 오른쪽 서브셋의 샘플 수입니다.

6.5 계산 복잡도

- 예측을 하려면 결정 트리를 루트 노드에서 리프 노드까지 탐색해야 합니다.
- 일반적으로 결정 트리를 탐색하기 위해서는 약 $O(log_2(m))$개의 노드를 거쳐야 합니다.
- 그래서 큰 훈련 세트를 다룰 때도 예측 속도가 매우 빠릅니다.

6.6 지니 불순도 또는 엔트로피

- 기본적으로 지니 불순도가 사용되지만 criterion 매개변수를 "entropy"로 지정해 엔트로피 불순도를 사용할 수 있습니다.

$$H_i = -\sum_{k=1}^{n} P_{i,k} log_2(P_i, k)$$

- 지니 불순도는 계산이 빠르기 때문에 기본값으로 좋습니다.
- 하지만 다른 트리가 만들어지면 지니 불손도가 가장 빈도 높은 클래스를 한쪽 가지로 고립시키는 경향이 있습니다.
- 엔트로피는 좀 더 균형잡힌 트리를 만듭니다.

6.7 규제 매개변수

- 결정 트리는 Train Data에 대한 제약사항이 거의 없습니다.
- 제한을 두지 않으면 트리가 Train Data에 아주 가깝게 맞추려고 해서 오버피팅될 가능성이 큽니다.

- 오버피팅을 피하기 위해 결정 트리의 자유도를 제한할 필요가 있는데 이를 규제라고 합니다.
- min_으로 시작하는 매개변수를 중가시키거나 max_로 시작하는 매개변수를 감소시키면 모델에 규제가 커집니다.

6.8 회귀

- 분류와 차이점은 각 노드에서 클래스를 예측하는 대신 어떤 값을 예측한다는 점입니다.
- CART 알고리즘은 훈련 세트를 불순도 최소화하는 방향으로 분할하는 대신 MSE를 최소화하도록 분할하는 것을 제외하고는 비슷하게 작동합니다.
- 회귀 작업에서도 결정 트리가 오버피팅되기 쉽습니다.

6.9 불안정성

- 결정 트리는 이해하고 해석하기 쉬우며, 사용하기 편하고, 여러 용도로 사용할 수 있으며, 성능도 뛰어납니다.
- 결정 트리는 계단 모양의 결정 경계를 만들기 때문에 훈련 세트의 회전에 민감합니다.
- 이런 문제를 해결하는 방법은 훈련 데이터를 더 좋은 방향으로 회전시키는 PCA 기법을 사용하는 겁니다.
- 결정 트리의 주된 문제는 Train Data에 있는 작은 변화에도 매우 민감하다는 것입니다.
- 랜덤포레스트는 많은 트리에서 만든 예측을 평균하여 이런 불안정성을 극복할 수 있습니다.