과제 1113

2315028 김성현

▼ 결정트리

▼ 의사결정트리(DT, decision tree) : 해결방법을 찾기위한 일련의 결정을 나타내는 구조 (질문의 연쇄를 통해 정답을 결정해나감)

기계학습에 사용되는 지도학습방법의 일종, 분류문제에 자주활용되지만 회귀문제도 풀 수 있음

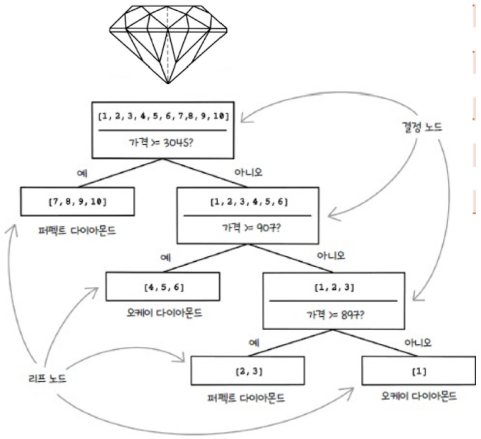
데이터를 필터링하는 질문을 생성. 패턴을 찾고, 정확히 필터링함

▼ ex_

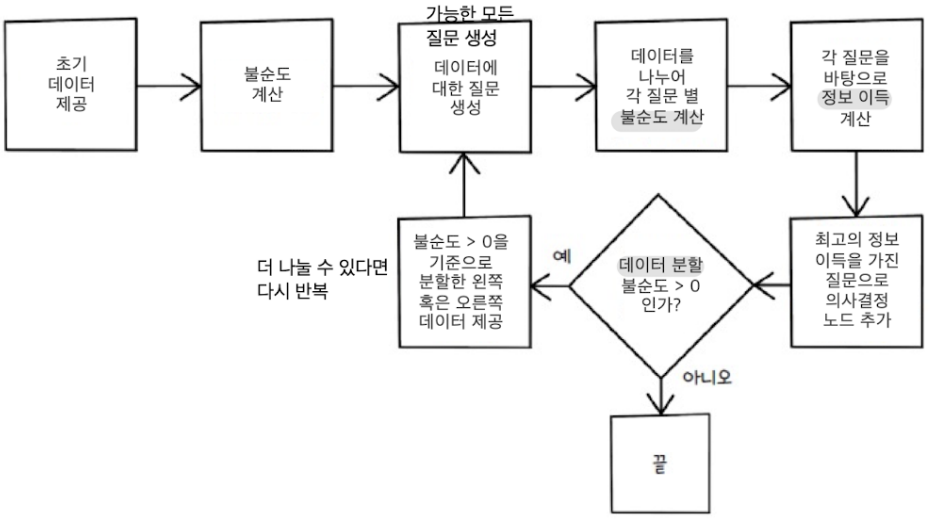
feature target

	캐럿	가격	컷
1	0.21	327	오케이
2	0.39	897	퍼펙트
3	0.50	1,122	퍼펙트
4	0.76	907	오케이
5	0.87	2,757	오케이
6	0.98	2,865	오케이
7	1.13	3,045	퍼펙트
8	1.34	3,914	퍼펙트
9	1.67	4,849	퍼펙트
10	1.81	5,688	퍼펙트

2가지의 클래스
(오케이 / 퍼펙트)

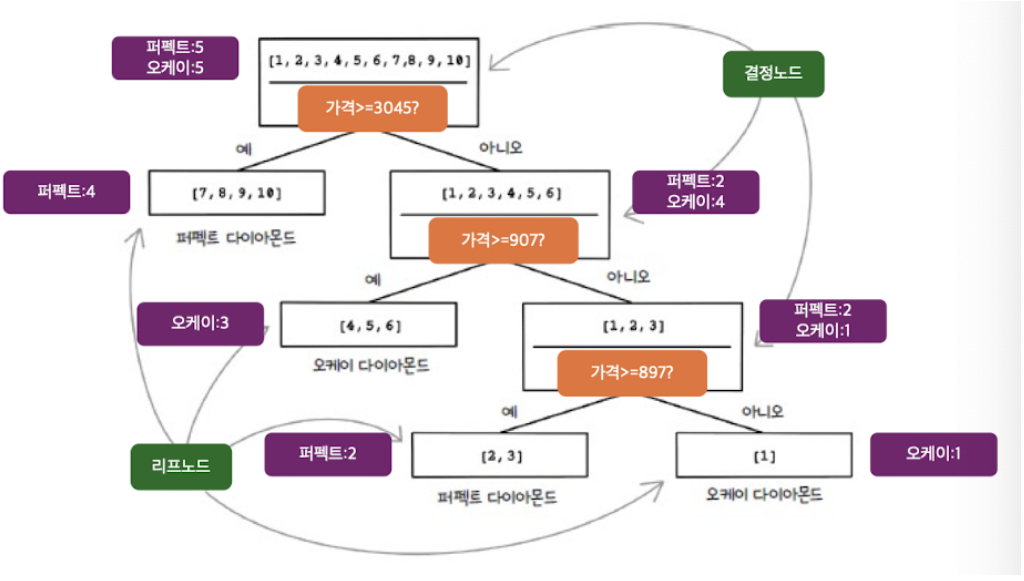


캐럿과 가격에 따라 다이아몬드의 컷을 어떻게 분류할 수 있을까?



▼ 질문

의사결정트리는 질문에 질문을 이어가며 데이터를 계속 필터링해 데이터를 분류함

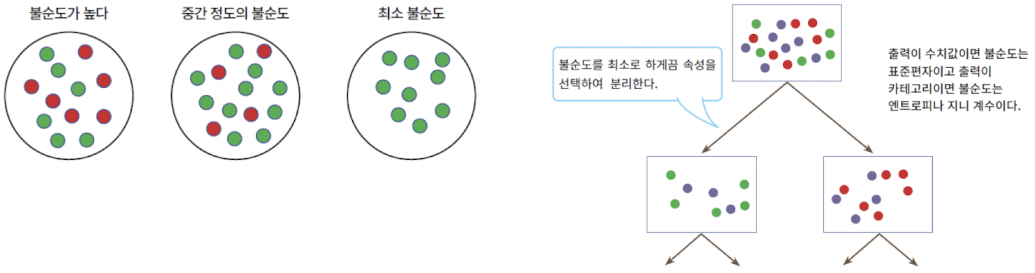


- 질문
  - 특징 (feature) : 질문에 포함된 특징
  - 값 (value) : 비교 대상보다 크거나 같은지 판단의 기준이 되는 상수 값

  ' 특징 x 값 ' 만큼의 질문을 할 수 있음
- 노드 트리
  - 결정 노드 (decision node) : 데이터셋을 분할 또는 필터링하는 노드. 질문과 항상 함께하는 노드
  - 리프 노드 (leaf node) : 데이터 목록만 가지는 노드. 분류가 완료된 상태인 노드
- 클래스 그룹화 맵

- 각 클래스 별 데이터 개수를 저장. ' 키 - 값 ' 형태로 이루어짐.
- 키(클래스), 값 (클래스에 속한 데이터의 수)
- 정보를 저장하는 이유 : 불순도를 계산하기위해서..

▼ 불순도



- 불순도(impurity) : 특정 데이터 내 데이터가 얼마나 다양한지 혹은 불확실한 상태를 가지는지 측정한 지표. 노드 안의 다른 종류의 데이터가 얼마나 섞여있는지를 나타냄.

  가장 적합한 질문을 선택하는데 사용함.

  불순도가 높다 - 여러 클래스가 균등하게 섞여있음을 의미

  불순도가 낮다 - 특정 클래스가 많은 경우를 의미

- 불순도 기반 질문

  불순도가 낮으면 변별력이 좋은 질문임

  의사결정트리는 변별력이 좋은 질문을 통해 자식노드의 불순도를 최소로 만들어가는 일종의 탐욕적인 알고리즘의 일종

- 불순도 지표

  ▼ 지니 계수 : 소득불평등을 측정하기위해 경제학자 지니가 만들었음.

  머신러닝에서는 클래스 불순도를 측정하기위해 사용

$$Gini = 1 - \sum_{i=1}^{c} p_i^2$$

  * P_i: 클래스 i 에 해당할 확률

  0~1 사이의 값으로 값이 클수록 불평등함을 의미

  - 특징

    계산이 비교적 단순해 속도 측면에서 유리.

분할 시 더 큰 그룹을 분리하려는 경향이 있음 → 얕고 간단한 트리 생성 가능성이 높음

CART(분류, 회귀 트리)의사결정트리 알고리즘에 활용됨

▼ 엔트로피 : 정보이론에서 등장한 이론

정보의 불확실성을 수치화하는데 활용함. 주어진 확률분포에서 정보의 무질서도를 측정함

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

$*$ P_i: 클래스 i 에 해당할 확률

- 특징

  계산이 복잡해 속도측면에서 불리함

  여러 클래스 간 섞임정도를 균형있게 고려함 → 깊고 복잡한 트리를 생성 가능성이 높음

  ID3, C4.5등의 의사결정 트리 알고리즘에 활용됨

▼ ex_ 불순도 계산

오케이 다이아몬드 5개 / 퍼펙트 다이아몬드 5개 ⇒ 총 다이아몬드 10개

$$Gini = 1 - \sum_{i=1}^{c} p_i^2$$

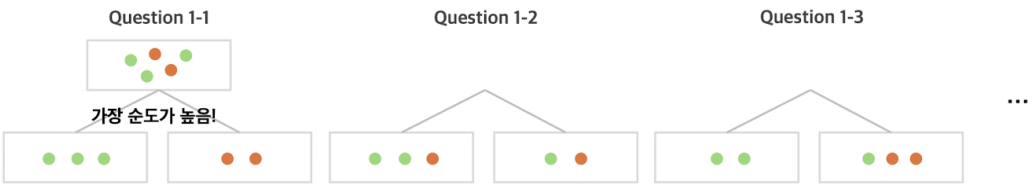
$$Gini = 1 - \left(\left(\frac{5}{10}\right)^2 + \left(\frac{5}{10}\right)^2\right) = 1 - \frac{50}{100} = 0.5$$

$$Entropy = -\sum_{i=1}^{n} p_i \log_2 p_i$$

$$Entropy = -\frac{5}{10}\log\frac{5}{10} - \frac{5}{10}\log\frac{5}{10} = -\log\frac{1}{2} = 0.301$$

식이 다르기에 두 결과는 다름!

▼ 정보 이득



의사결정 트리는 질문을 통해 데이터를 분류해 정보 이득이 가장 큰 질문을 선택함

부모노드의 불순도와 자식노드들의 불순도의 차이를 통해 정보이득을 계산

> 정보이득(information gain) = (부모) 불순도 - (자식들) 가중 평균 불순도

자식노드들의 가중평균 불순도 = 해당 자식노드 데이터의 수 / 부모노드 데이터의 수

→ 정보이득을 통해 불순도가 낮은 경우를 선택함

▼ 불순도 vs 정보이득

불순도 : 노드간의 불순도

정보이득 : 부모 불순도와 자식 가중평균 불순도의 차이

▼ 알고리즘

```
Function decision_tree(samples):
    impurity = calculate_impurity(samples)  불순도 계산

    if impurity == 0:
        return LeafNode(samples)  더이상 나눌 필요없다면 리프노드로
    else:
        questions = generate_possible_questions(samples)  가능한 질문들을 만듬

        best_gain = 0
        best_question = null
        best_splits = null

        for each question in questions: 질문을 하나씩 살핌
            true_data, false_data = split_data(samples, question)  데이터 분리

            if true_samples is empty or false_samples is empty:
                continue

            true_impurity = calculate_impurity(true_samples)
            false_impurity = calculate_impurity(false_samples)  분리 된 그룹들 불순도 계산

 정보이득 계산 gain = calculate_information_gain(impurity, true_samples, false_samples)

            if gain > best_gain:
                best_gain = gain           최고의 정보이득 가진 질문을 찾음
                best_question = question
                best_splits = (true_samples, false_samples)

        decision_node = DecisionNode(best_question)  질문을 의사결정 노드에 추가

        decision_node.branch_true = decision_tree(best_splits[0])
        decision_node.branch_false = decision_tree(best_splits[1])  재귀적으로 반복하게됨

        return decision_node
```

1. 초기 데이터 불순도 계산
2. 질문생성

   모든 feature과 모든 value를 순회하며 가능한 모든 질문을 생성
3. 데이터 분리 후 불순도 계산

   질문별 데이터 분리 → 분리된 그룹들 불순도 계산
4. 정보이득 계산

5. 최고의 정보이득 가진 질문으로 의사결정 노드 추가
6. 재귀적 반복

   불순도가 0 이 될때까지

- 평가 및 성능측정

  ▼ 혼동행렬 (confusion matrix)

  : 모델이 예측한 결과와 실제 값을 비교하는 성능 평가 도구

	예측한 양성	예측한 음성	
실제 양성	참 양성 TP	거짓 음성 FN	민감도(sensitivity) TP / TP+FN
실제 음성	거짓 양성 FP	참 음성 TN	특이도(specificity) TN / TN+FP
	정밀도(precision) TP / TP+FP	음의 정밀도 (negative precision) TN / TN+FN	정확도(accuracy) $\frac{TP+TN}{TP+TN+FP+FN}$

  - 정확도 : 전체 데이터 중에서 올바르게 예측된 비율

    예측 = 실제인 경우, TP, TN

    일반적으로 가장 많이 활용되는 지표. 상황에 따라 좋은지표가 아닐 수 있음

    단순히 예측한 비율만 나타내기에.. 클래스의 양성/ 음성 비율을 고려하지 않음

    → 데이터 내 클래스가 불균형할때 성능을 잘못평가할 수 있음
  - 민감도 : (=재현율) 실제로 양성인 데이터 중에서 올바르게 긍정으로 예측한 비율

    거짓 부정을 최소화하려는 상황에서 유용함

    놓치지 말아야할 것이 중요할 경우에 사용
  - 특이도 : 실제로 음성인 데이터 중에서 올바르게 부정으로 예측한 비율

    음성인 데이터를 양성으로 잘못예측하면 큰 문제가되는 경우에 유용함
  - 정밀도 : 긍정으로 예측한 데이터 중에서 실제로 긍정인 비율

    정확성이 중요하거나 실수 비용이 큰 경우

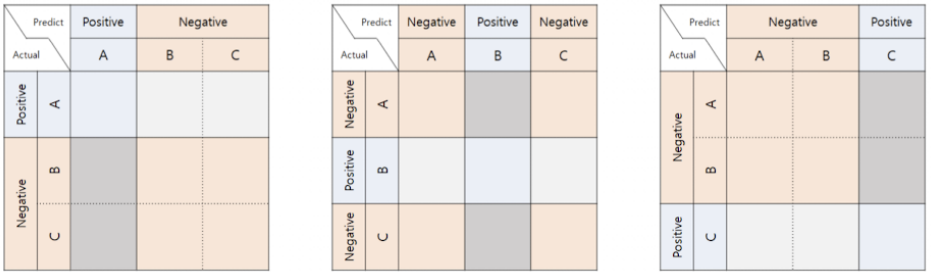
    거짓 긍정을 최소화하려는 상황에서 유용

  +) 혼합지표 : F1Score(정밀도와 재현율의 조화평균), AUC(민감도와 특이도 활용)

  ▼ ex_

			실제	예측	실제 = 예측
	캐릿	가격	컷	예측	
1	0.26	689	오케이	오케이	✓
2	0.41	880	퍼펙트	퍼펙트	✓
3	0.52	1,012	퍼펙트	퍼펙트	✓
4	0.76	907	오케이	오케이	✓
5	0.81	2,650	오케이	오케이	✓
6	0.90	2,634	오케이	오케이	✓
7	1.24	2,999	퍼펙트	오케이	†
8	1.42	3,850	퍼펙트	오케이	†
9	1.61	4,345	퍼펙트	퍼펙트	✓
10	1.78	3,100	오케이	퍼펙트	†

	예측한 양성	예측한 음성	
실제 양성	참 양성 4	거짓 음성 1	민감도(sensitivity) 4 / 4+1 = 0.8
실제 음성	거짓 양성 2	참 음성 3	특이도(specificity) 3 / 3+2 = 0.6
	정밀도(precision) 4 / 6 = 0.67	음의 정밀도 (negative precision) 3 / 4 = 0.75	정확도(accuracy) $\frac{7}{10}$ = 0.7



클래스가 여러개인 경우에, 평가지표들은 다중 클래스 상황에 맞게 확장됨

▼ 개별트리 모델의 한계점

1. 과적합의 위험이 높음 → 데이터 변화에 민감함 (결과 안정성이 떨어짐)
2. 복잡한 비선형 경계를 학습하기 어려움 → 고차원 데이터를 다룰때 불리
3. 계층적구조 → 오류에 민감함 (오류전파)
4. 스케일이나 범위가 큰 특성을 중요하게 취급하는 경향이 있음

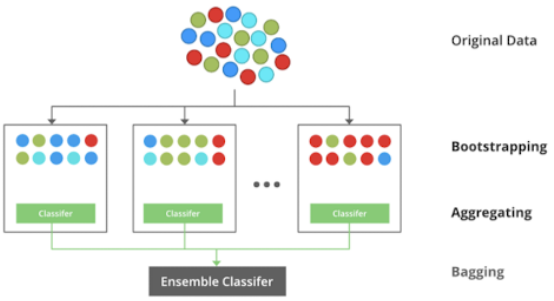
▼ 트리기반 앙상블 모델

한계점을 극복하고자, 트리 여러개를 합쳐 결과를 내는 방식인 '앙상블'이 제안됨

▼ 배깅 계열

: 특징과 데이터를 무작위로 샘플링하는 것을 반복해 다양한 의사결정트리를 만든 후 결과를 총합하는 방식. 병렬적 학습방식

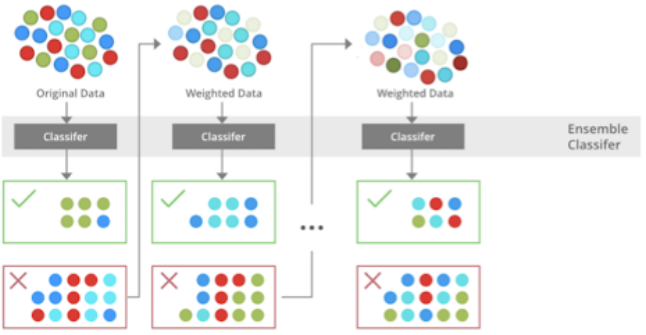
ex_ random forest



▼ 부스팅 계열

: 약한 트리의 가중치를 지속적으로 업데이트해 점차 강한트리로 연속적으로 업데이트해나가는 방식. 순차적 학습방식

ex_ Xgboost, LightGBM, Catboost



그외) 보팅 : 여러개의 서로다른 모델의 예측결과를 투표해 최종예측을 결정하는 방식

스태킹 : 여러 모델의 예측을 메타모델의 입력으로 사용해 최종 예측을 만드는 방식