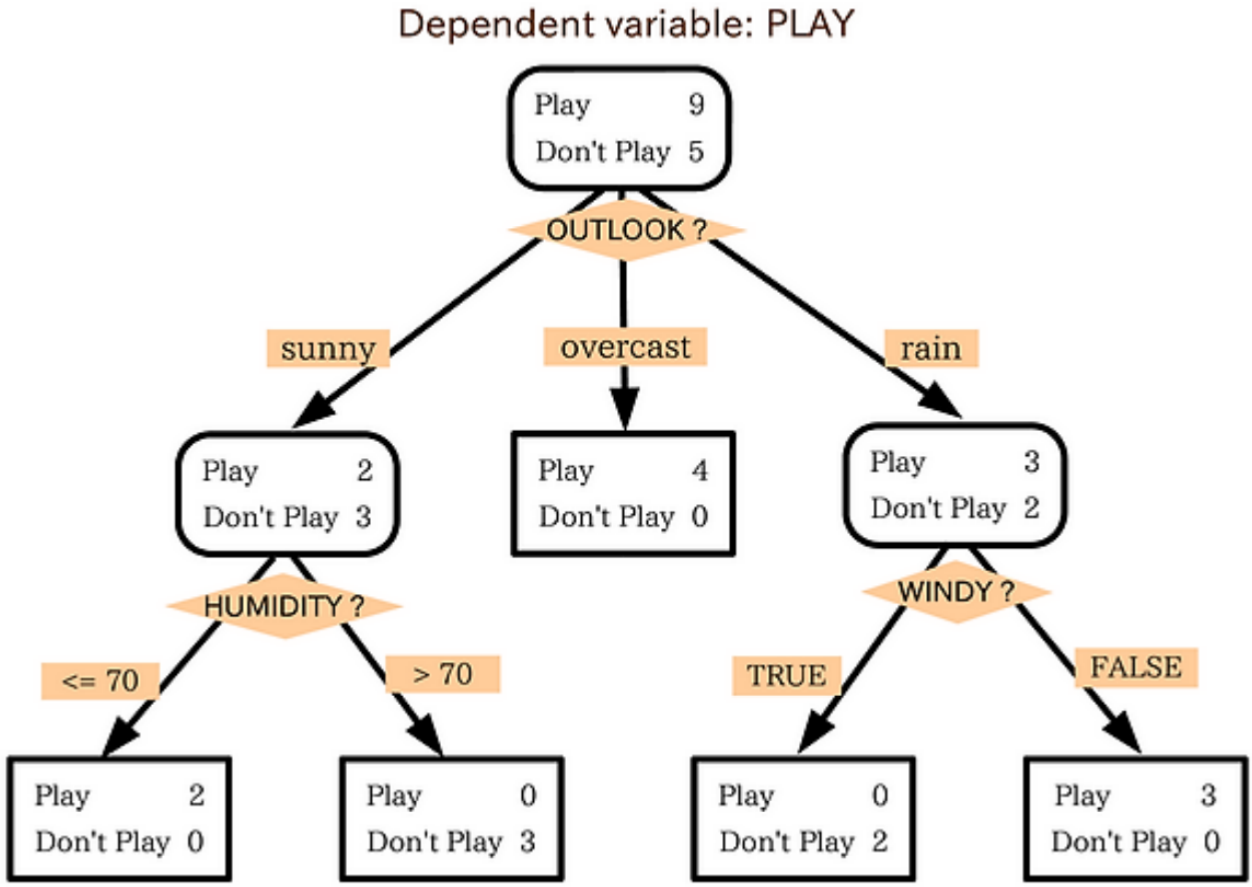
# Decision Tree(의사결정 나무)

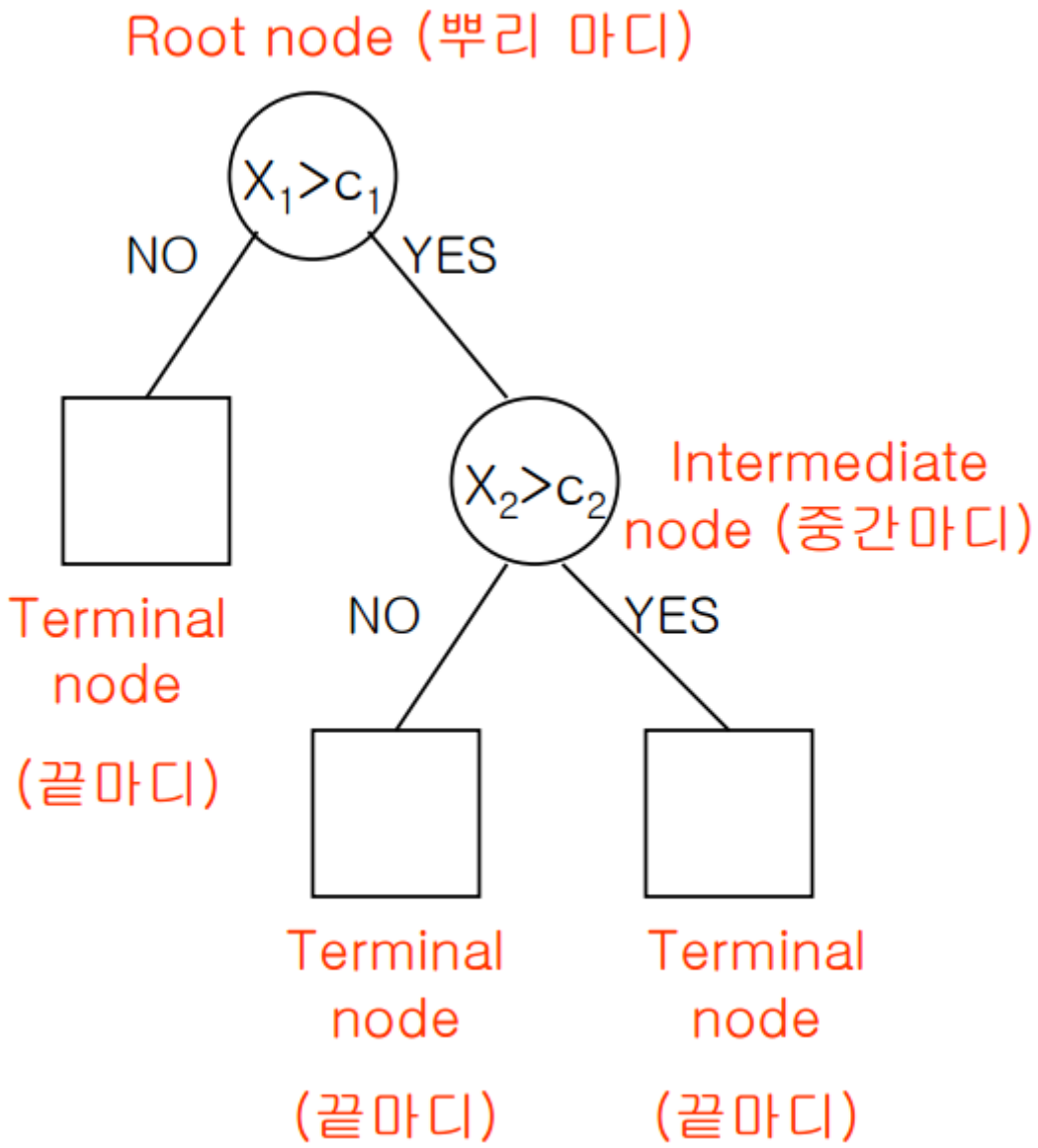
## 1.설명

의사결정 나무는 데이터를 분석하여 이들 사이에 존재하는 패턴을 예측 가능한 규칙들의 조합으로 나타내며 그 모양이 '나무'같아서 의사결정 나무로 불리는 것으로 질문을 던져서 대상을 좁혀나가는 '스무고개'와 비슷한 개념이다. 분류(Classification)과 회귀(Regression)에 모두 사용될 수 있다.




출처 : https://imgur.com/ZKDnzOB

위의 이미지는 운동 경기가 열리는 경우를 조사한것으로 Play값의 이진(0,1)에 따라 분류한것이다. 날이 맑은 경우 습도가 70 이하면 운동경기가 열렸고 70 이상일 경우 운동경기가 열리지 않았다.




출처 : http://imgur.com/EBKl1I3

크게 위와 같은 유형으로 동작하며 초기지점(root node)에서 분기가 거듭될 수록 해당 데이터의 갯수는 줄어든다. 각 terminal node의 합은 root node의 데이터 수와 일치한다. 범주 형의 데이터 경우 위의 운동경기 열리는 경우 예측처럼 다양한 조건과 기준 값에 따른 범주 예측이 가능하고, 연속형 데이터의 수치의 경우 평균과 표준편차에 기초하여 분류가 가능하다. 의사결정 나무의 대표값은 분류일 경우 최빈값이 연속형 데이터일 경우 평균 값이 된다.

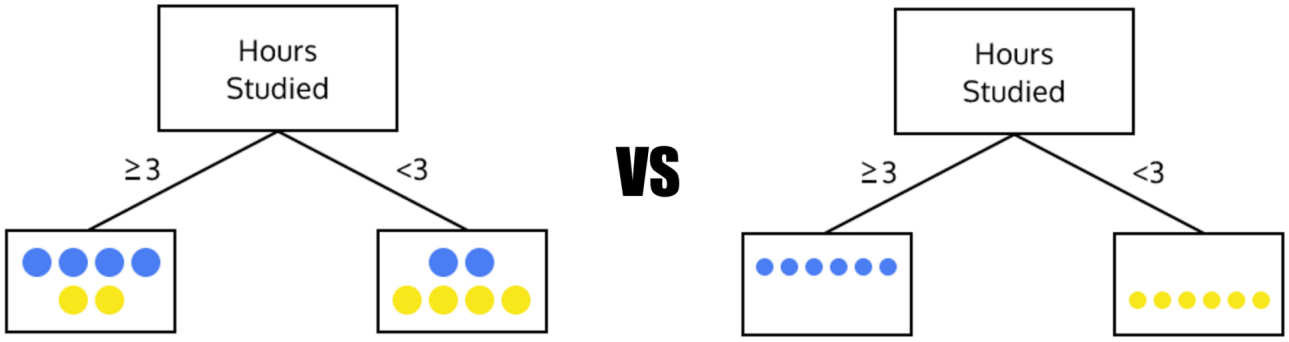
## 1-2.의사결정 나무 분류 학습법

1. 여러가지 독립 변수 중 하나의 독립 변수를 선택, 해당 독립변수에 대한 기준 값(threshold-분류 규칙)을 정한다.(X1 > C1)
2. 전체 학습 데이터 집합(root node)을 독립 변수의 기준값에 따라 분류하여 자식노드로 분류한다.
3. 1~2를 반복하여 하위의 자식노드를 만들어 나간다. 단 자식노드에 한가지 클래스의 데이터만 존재하면 중지.

# 2.특징

## 2-1.순도 (Purity)

의사결정나무에 따르면 기준값을 통해 두가지로 데이터를 구분한다.




출처 : [https://hleecaster.com/ml-decision-tree-concept/

왼쪽 그림처럼 데이터의 분할이 깔끔하지 않은 것을 순도가 낮다고 할 수 있다. 이처럼 순도를 표현하기 위해 **지니 불순도(Gini Impurity)**를 이용한다.

```
1 - '전체 데이터 중 각 레이블이 차지하는 개수의 비율의 제곱'
```

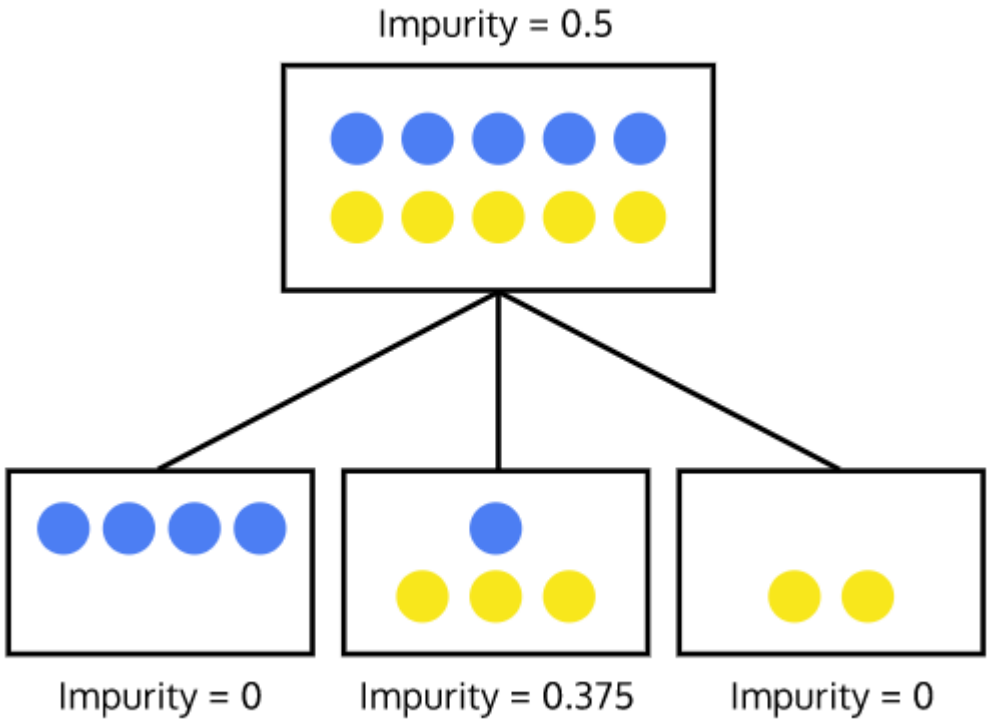
왼쪽의 그림의 데이터 순도는

```
1 - (2/6)^2 - (4/6)^2  = 0.45
```

지니 불순도는 순도가 높을 수록 0에 가까워진다. 지니 불순도가 작을수록 잘 분할된 데이터이다.

## 2-2. 정보 획득량 (Information Gain)

지니 불순도 계산을 통해 정보획득량도 계산할 수 있다.



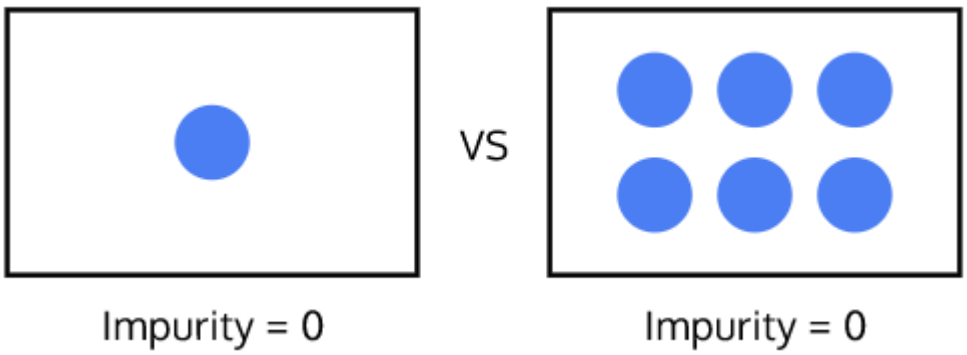

출처 : [https://hleecaster.com/ml-decision-tree-concept/

위처럼 불순도 0.5의 데이터 세트를 각각 0, 0.375, 0 3개의 데이터 세트로 분할 했을 때, 얻은 정보획득량은 아래와 같이 계산할 수 있다.

```
상위 Node Gini Impurity - (각 데이터 세트 Gini Impurity)
```

```
0.5 - (0 + 0.375 + 0) = 0.125
```

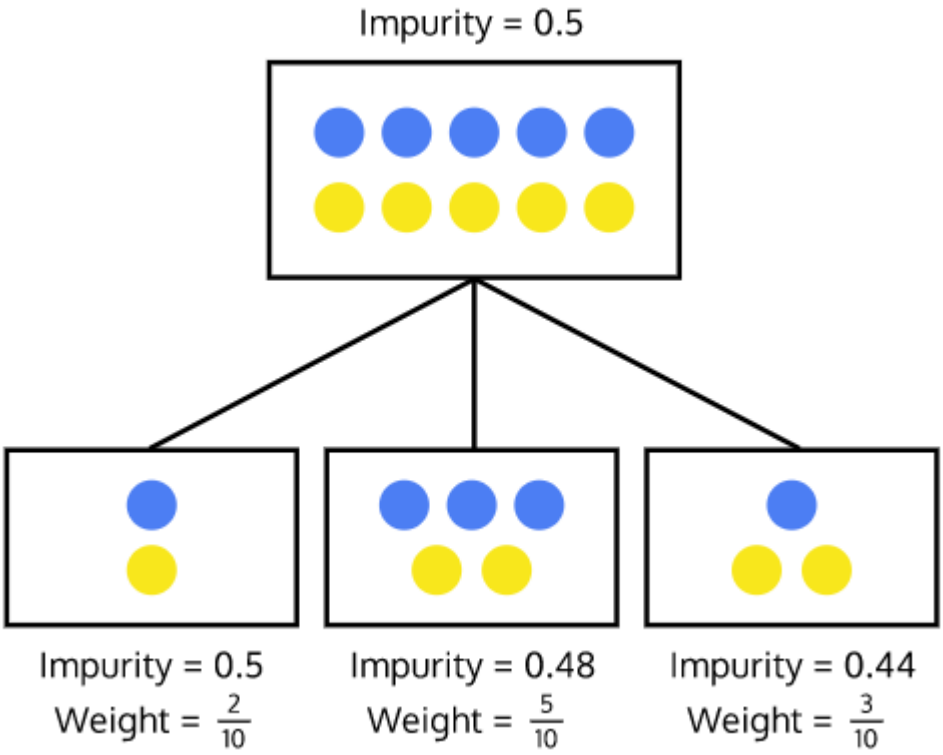
**분할된 데이터 세트들의 분순도가 작을수록 정보 획득량이 증가한다.**




출처 : [https://hleecaster.com/ml-decision-tree-concept/

다만 위와 같이 불순도는 모두 0이지만 오른쪽 데이터 세트가 데이터 갯수가 충분히 많고 분류가 우연이 아니라고 볼수 있으므로 더 의미가 있다.

**생성된 데이터 세트의 크기에 따라 가중치가 적용된 정보획득량을 계산하여 실제 의미있는 정보 획득(Weighted Information Gain)을 계산할수 있다.**




출처 : [https://hleecaster.com/ml-decision-tree-concept/
위 그림처럼 분할 후 생성된 데이터의 크기(비율)에 따라 가중치를 구해놓고 불순도에 곱해서 정보 획득량을 구한다.

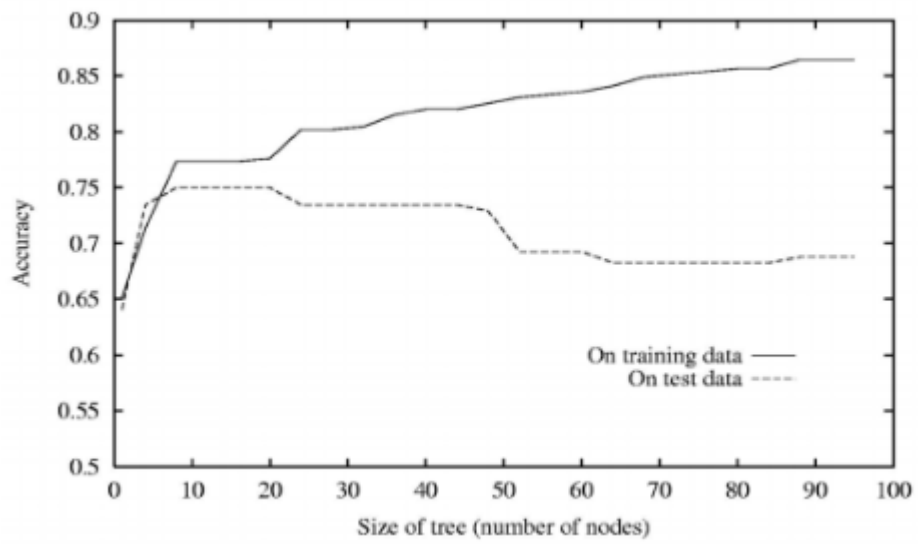
```
0.5 - ((2/10)*0.5 + (5/10)*0.48 + (3/10)*0.44) = 0.026
```

**데이터 세트의 크기가 작을 수록 불순도의 영향력이 작아진다.**

**정보 획득량이 큰 순서대로 질문 (스무고개시 큰 질문 먼저)하는 것이 중요하다.**

## 2-3 가지치기

의사결정 나무의 경우 분기수 증가에 따라 정확도가 증가하다가 일정수준 이상(과적합)될 경우 오히려 오분류율이 증가한다.




**출처 : [https://sanghyu.tistory.com]**

학습용 데이터에 최적화 되어 일반화 되지 못해서 발생한 문제이므로, 가지치기(Pruning)을 통해 오분류율이 증가하는 시점 적절히 가지를 쳐주어야한다.

1. 테스트 데이터를 재귀적 돌려 특정 가지를 제거후 정확도가 올라가면 해당 가지치기 수행
2. 비용함수 이용 (CC(T)=Err(T)+α×L(T))

   CC(T)=의사결정나무의 비용 복잡도(=오류가 적으면서 terminal node 수가 적은 단순한 모델일 수록 작은 값)

   ERR(T)=검증데이터에 대한 오분류율

   L(T)=terminal node의 수(구조의 복잡도)

   Alpha=ERR(T)와 L(T)를 결합하는 가중치(사용자에 의해 부여됨, 보통 0.01~0.1의 값을 씀)

---

## 2-3. 장단점

- 장점 : 대용량 데이터에서도 빠르게 생성 가능, 해석이 용이, 모델의 시각화에 용이, 정규화나 표준화 등의 전처리가 필요없다.
- 단점 : 과적합으로 정확도가 떨어진다. 계속 분류조건을 추가하게되면 실제 상황에서 유연하게 대처하는 능력이 떨어진다(새로운 데이터 예측 못함).

## 적합한 사례

- 신용평가 문제(대출 승인에 대한 의사결정 과정( 고객 신용평가 지수 모형 구성))
- 증상에 따른 환자의 병 확인
- 고객을 세분화한 후 목표 고객 선별하여 적절한 광고, 캠페인 진행
- 제품 선호 고객 예측