결정 트리
(Decision Tree)

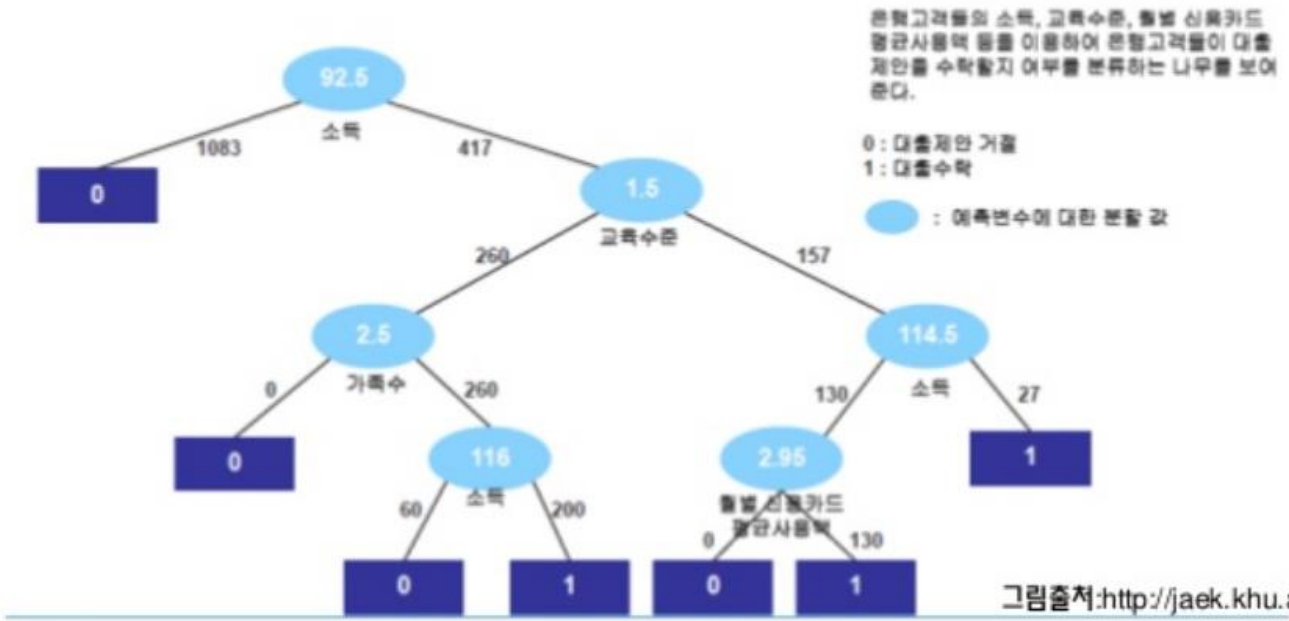
결정트리

동양미래대학교

정의

- ✓ 의사결정 규칙 (Decision Tree)을 도표화하여 관심대상이 되는 집단을 몇 개의 소집단으로 분류 (Classification)하거나 예측 (Prediction)을 수행하는 계량적 분석 방법

- ✓ 장점
  - 분석결과는 '조건 A이고 조건 B이면 결과집단 C'라는 형태의 규칙으로 표현되므로 이해가 쉽고, 분류 또는 예측을 목적으로 하는 다른 계량적분석 방법에 비해 쉽게 이해하고 활용 할 수 있음



그림출처:http://jaek.khu.ac.kr/datamining/684

의사결정 트리의 예

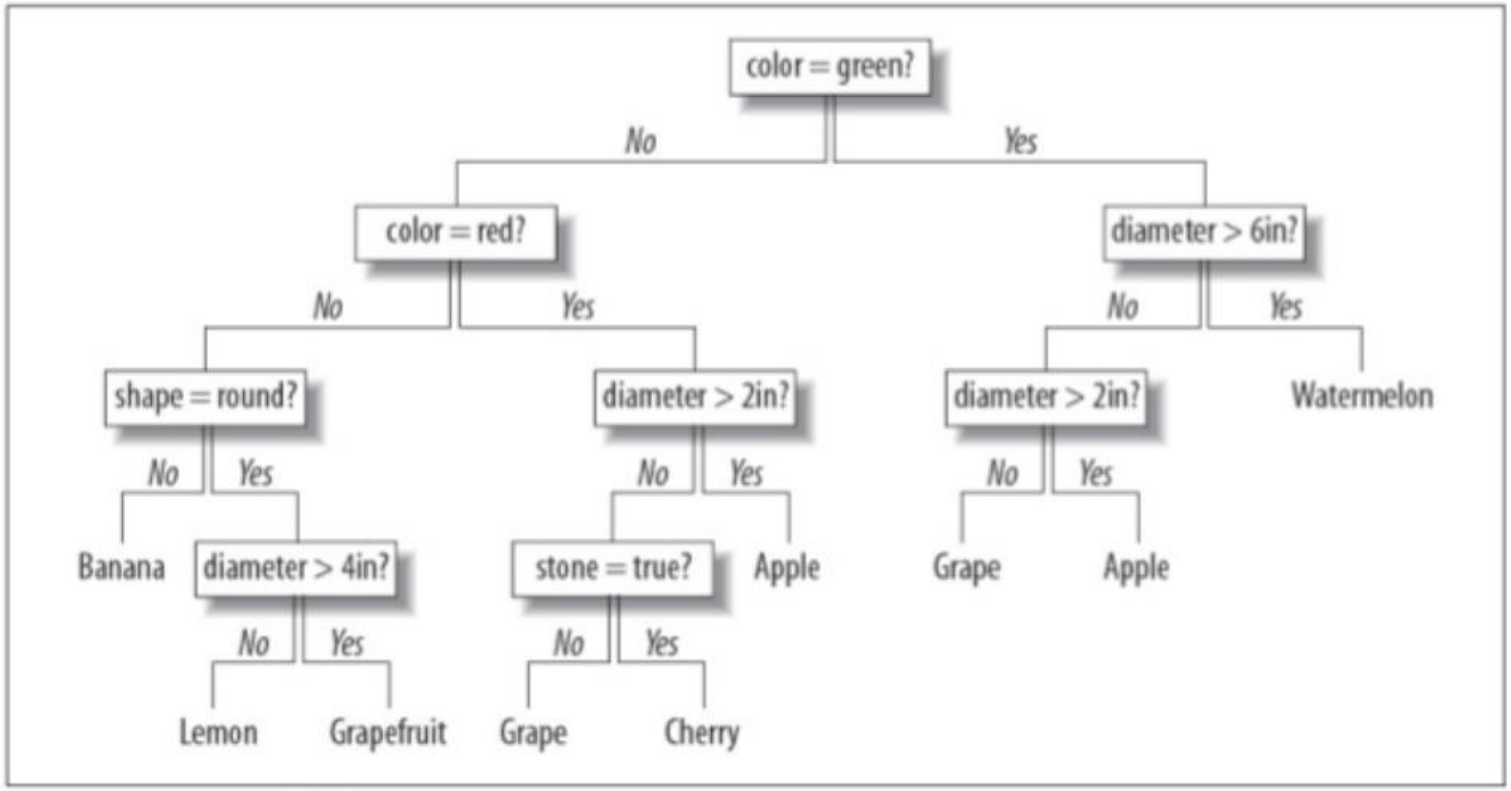
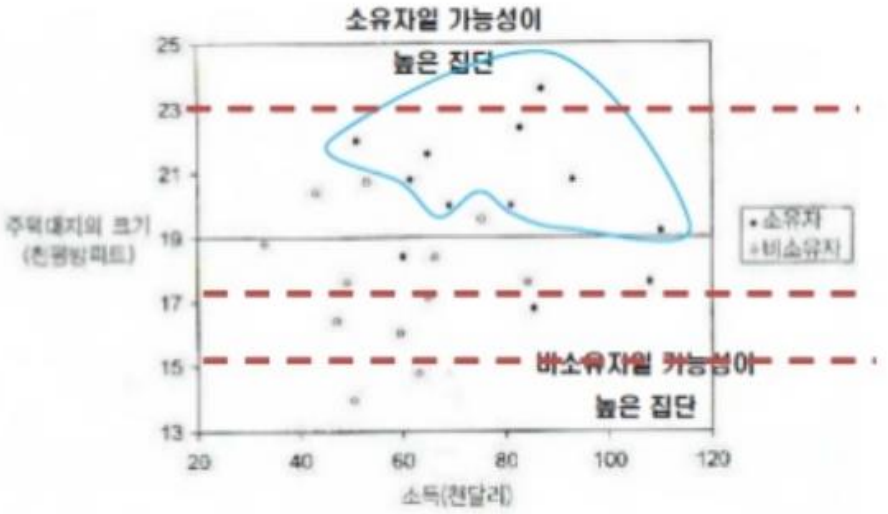


Figure 7-1. Example decision tree

의사결정 트리의 분할 속성 선택

- ✓ 분할된 데이터의 불순도를 얼마나 많이 제거 했는가로 속성과 속성값을 결정
  - 속성선택 : 현재의 불순도 – 노드를 분리한 다음의 불순도
- ✓ 예 : 승차식 잔디깍기의 구매여부 판단

가구일련번호	소득(천달러)	주택대지의 크기(천평방피트)	승차식 잔디깎기 기계의 소유여부
1	60.0	18.4	소유자
2	85.5	16.8	소유자
3	64.8	21.6	소유자
4	61.5	20.8	소유자
5	87.0	23.6	소유자
6	110.1	19.2	소유자
7	108.0	17.6	소유자
8	82.8	22.4	소유자
9	69.0	20.0	소유자
10	93.0	20.8	소유자
11	51.0	22.0	소유자
12	81.0	20.0	소유자
13	75.0	19.6	비소유자
14	52.8	20.8	비소유자
15	64.8	17.2	비소유자
16	43.2	20.4	비소유자
17	84.0	17.6	비소유자
18	49.2	17.6	비소유자
19	59.4	16.0	비소유자
20	66.0	18.4	비소유자
21	47.4	16.4	비소유자
22	33.0	18.8	비소유자
23	51.0	14.0	비소유자
24	63.0	14.8	비소유자



주택대지 크기속성의 분할 값 선택
- 15을 선택했을때 불순도 ?
- 17을 선택했을때 불순도?
- 19? 21? 23?을 선택했을 때는???

소득액 속성의 분할 값 선택
- 40을 선택했을때 불순도 ?
- 60을 선택했을때 불순도?
- 80? 100?을 선택했을 때는???

의사결정 트리의 분할 속성 선택

- ✓ 어떤 입력변수를 이용하여 어떻게 분리하는 것이 목표변수의 분포를 가장 잘 구별해 주는지를 파악하여 자식마디가 형성되는데, 목표변수의 분포를 구별하는 정도를 순수도(Purity), 또는 불순도(Impurity)에 의해서 측정
  - 순수도 (Purity) : 특정 범주의 개체들이 포함되어 잇는 정도를 의미한다.
  - 불순도(impurity) : 얼마나 다양한 범주들의 개체들이 포함되어있는 가를 의미

- ✓ 분할속성의 선택
  - 부모마디의 순수도에 비해서 자식마디들의 순수도가 증가하도록 자식마디를 형성
    - 예를 들어 그룹0과 그룹 1의 비율이 45%와 55%인 마디는 각 그룹의 비율이 90%와 10%인 마디에 비하여 순수도가 낮다 (또는 불순도가 높다)라고 이야기 한다.

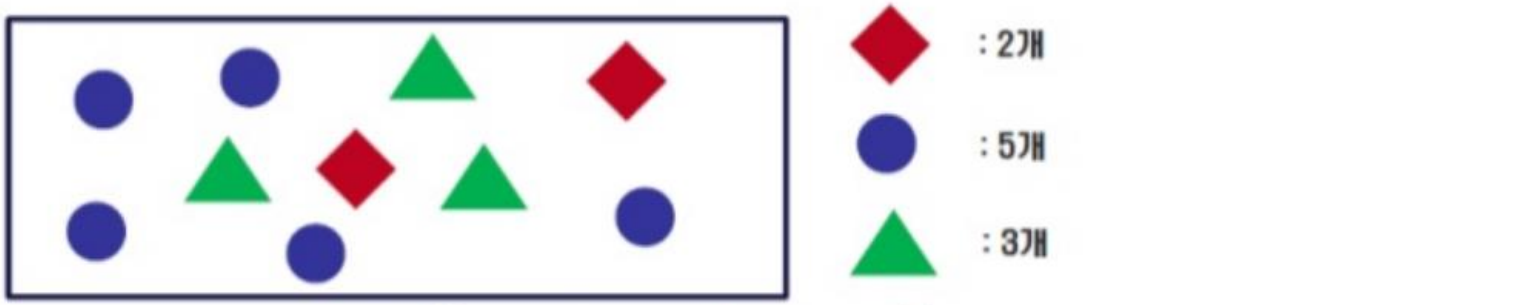
- ✓ 불순도의 측정
  - 카이제곱 통계량의 P값
  - 지니 지수 (Gini Index)
  - 엔트로피 지수(Entropy Index)

지니 지수 (Gini Index):

- ✓ 불순도를 측정하는 하나의 지수로서 지니지수를 가장 감소시켜주는 예측변수와 그 때의 최적 분리에 의해서 자식마디를 선택

$$I(A) = 1 - \sum_{1}^{m} p_k^{\,2}$$

Pk : 직사각형 A에서 K 집단에 속하는 관측치비율

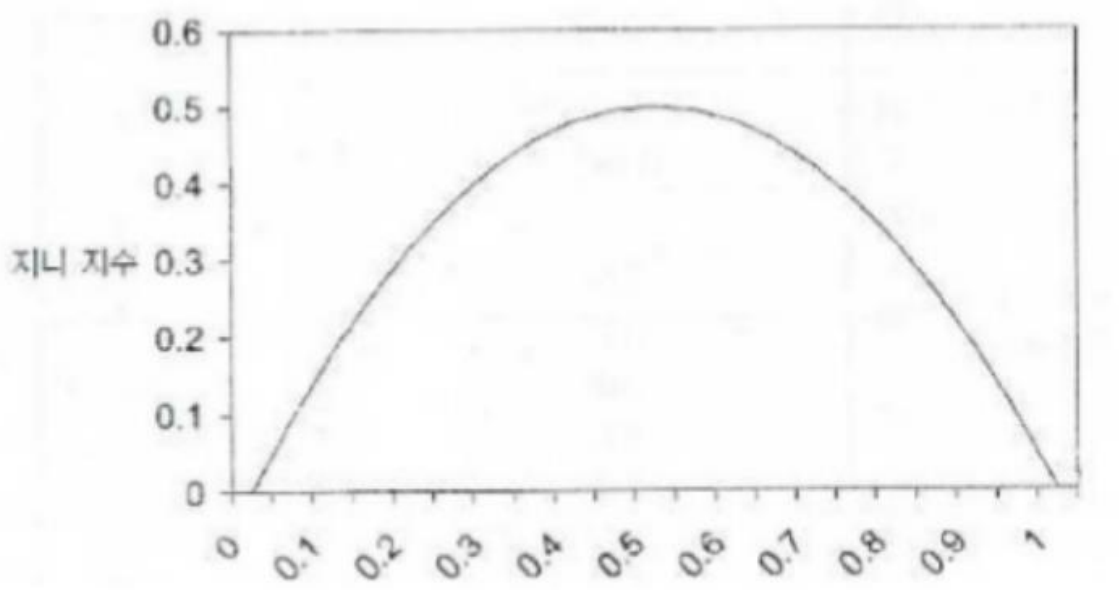


$$I(A) = 1 - \sum_{1}^{m} p_k^{\,2} = 1 - \left[\left(\frac{2}{10}\right)^2 + \left(\frac{3}{10}\right)^2 + \left(\frac{5}{10}\right)^2 \right] = 1 - (0.04 + 0.09 + 0.25)$$

❖ 지니 지수 (Gini Index)의 값 다이어그램

✓ 두개의 범주개체가 50대 50으로 구성될때 최대의 불순도값 0.5

지니 불순도 지수는 0~(m-1)/m 사이의 값을 갖음

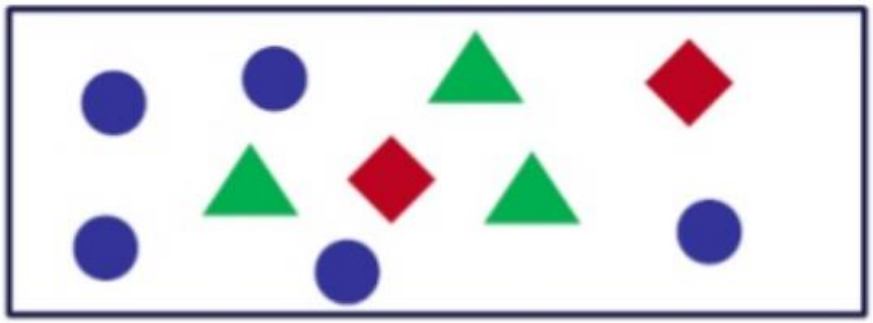


집단 1에 속하는 관찰치들의 비율의 함수로서 이진분류에 대한 지니 지수의 값

❖ 엔트로피 지수(entropy index)

엔트로피 지수 $entropy(A) = -\sum_{k=1}^{m} p_k \log_2(p_k)$

Pk : 직사각형 A에서 K 집단에 속하는 관측치비율



$$entropy\ (A) = -\sum_{k=1}^{m} p_k \log_2(p_k) = -\left[0.2 * \log_2(0.2) + 0.5 * \log_2(0.5) + 0.3 * \log_2(0.3)\right] = 1.4854$$

1. ID3 – 결정트리 알고리즘

동양미래대학교

한 번에 하나의 속성으로 데이터 분류하기

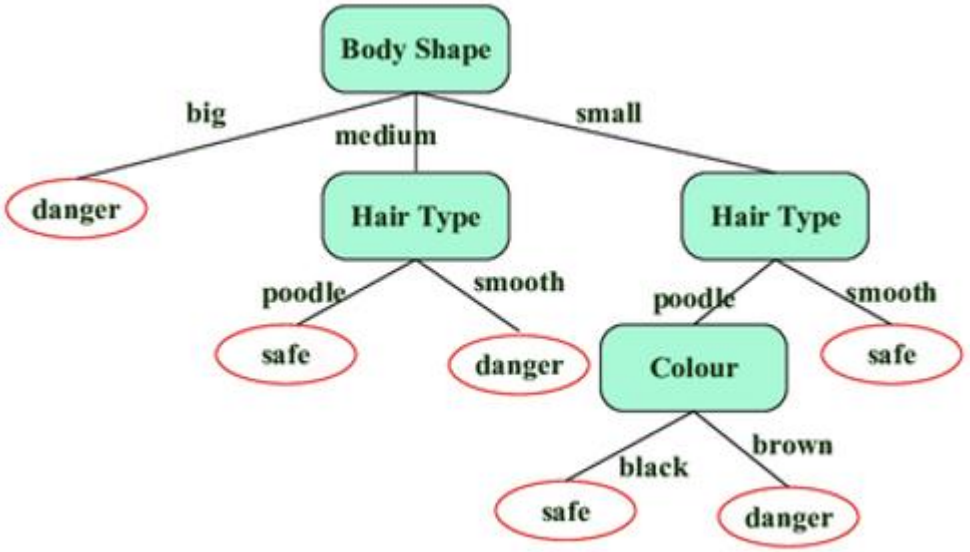
어떻게 분류 기준(속성)을 선택할까?

- 모든 속성에 대하여 정보이득을 계산하여 결정
  - 정보이득 계산 방법 : entropy, gini 방식
- 각 노드별로 위와 같이 정보이득 계산값으로 선택된 속성 선택

정보이득 계산(entropy)

- 분류하기 전과 분류 후의 변화량
- 계산값이 가장 높은 값을 선택 (변화량이 큰것)

Body Shape	Hair Type	Colour	safety
Big	Poodle	Black	safe
Big	Smooth	Black	Danger
Small	Poodle	Brown	Safe
medium	poodle	black	safe



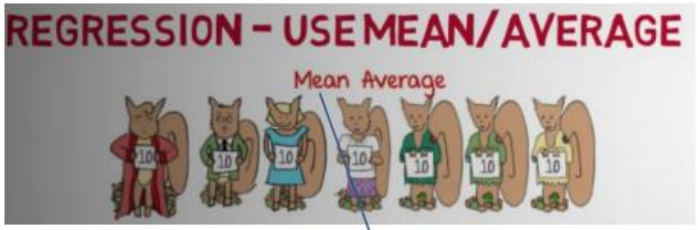
2. CART - 결정 트리 알고리즘

Classification And Regression Trees, 트리 기반 회기

Classification vs Regression Tree 비교

- Classification : 카테고리컬 변수에 사용
- Regression : 연속형 변수에 사용

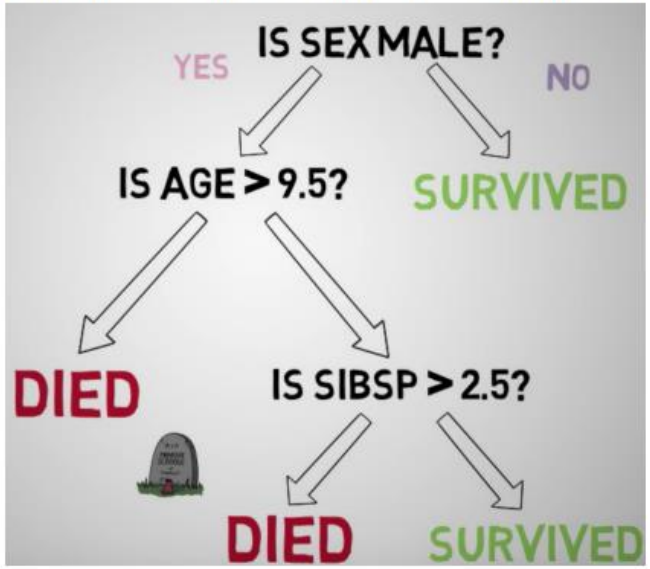
- 가장 많이 있는 속성을 기준으로 분류
- 성적 중 C가 가장 많다면, C인지 아닌지?



- 수치형 값의 평균으로 분류
- 선형으로 수치값을 분류할 때.

함수(알고리즘)을 어떻게 트리로 표현할까?

- 많이 사용하는 타이타닉 생존자를 분류하는 예시를 보자
- 수치형 값을 기준으로 3개의 알고리즘(함수)으로 데이터를 분류



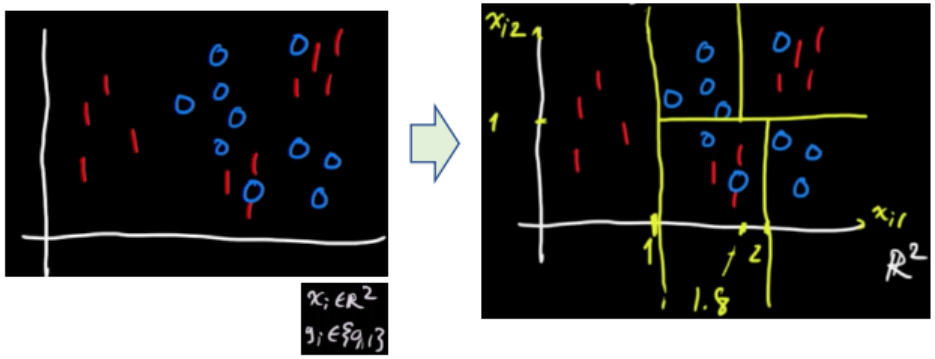
https://www.youtube.com/watch?v=DCZ3tsQIoGU

2. CART - 결정 트리 알고리즘

각 leaf node의 학습 에러(불순도)를 최소화 하는 Binary Tre

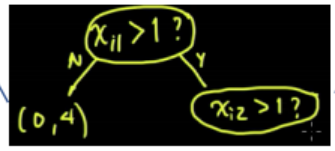
Classification Tree 예시

- 아래와 같이 임의 데이터를 생성
- Decision Tree는 데이터를 2개로 분할(Binary split)한다.
  - 이때 분류된 노드의 에러(불순도)를 최소화하는 분류기준을 선택



- 데이터를 binary로 분류하면, 아래과 같은 트리가 생성됨

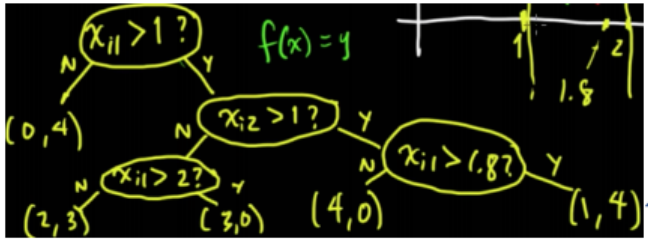
불순도 = 0
모두 1로 구성됨
더 이상 분류할 필요없음



불순도 = 0:10개, 1:7개
불순도를 최소화하기 위해
데이터 분류

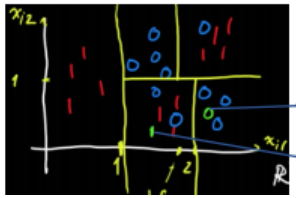
분류 알고리즘을 트리로 표현

- 그런데 특정 leaf node는 불순도가 높은 것이 보인다
- 이런 leaf에서는 어떻게 데이터를 분류할까?
  - → 다수결을 이용하여 분류
  - → 예를 들어 (2,3)은 0:2개, 1:3개이므로 1로 분류함



(1,4)은 다수결로 1로 분류함

그럼 새로운 데이터가 입력되면 어떻게 분류할까?



(3,0)은 0이 3개이므로 "0"으로 분류

(2,3)은 0이 2개, 1이 3개 이므로 "1" 로 분류

ID3와 CART 비교

주요 특징 비교

	ID3	CART
분류기준	Shannon entropy Information gain	Gini Index
분류 방식	• Field별 값을 기준으로 분할 • Grade란는 field에 (A, B, C)라는 값이 있으면, 각 속성별로 데이터를 분류함 (총 3번)	• 이진 트리를 사용하여, • 1개의 field의 값을 특정 기준값을 기준으로 • 크면 right, 작으면 left로 분류
지원 모델	Classfication	Classification + Rregression
과접합 방지	X	O (Leaf 데이터 갯수, 최소 변화량)
트리	트리	이진트리
회기모델 지원 (수치 데이터)	X	O
장점	구현 용이	부등호 질의가능 사후 가지치기 가능 (Leaf Node 통합) 데이터 해석이 용이(설명력)
단점	수치형 속성 사용불가 카테코리 속성이 많은 경우, Tree가 깊어짐 (가지가 많아지기 때문)	학습데이터가 충분해야 함 (과적합 유발) → 배깅/부스팅 활용

- **Entropy**
  - **예를 들어, 동물을 분류할때 "포유류인가?"라는 질문으로 먼저 분류하고,**
  - **"원숭이인가?" 라는 구체적인 질문으로 분류하는 것이 효과적 → 정보이득이 높음**
- **Gini**
  - **분류된 노드의 데이터 불순도 측정 (얼마나 많은 종류가 있는가?)**
  - **만약 노드에 1가지 종류의 데이터만 있다면, Purity(순수) 상태**
  - **따라서 불순도를 낮추는 것이 최적의 분류를 위한 feature를 찾는 것**

결정 트리의 장단점

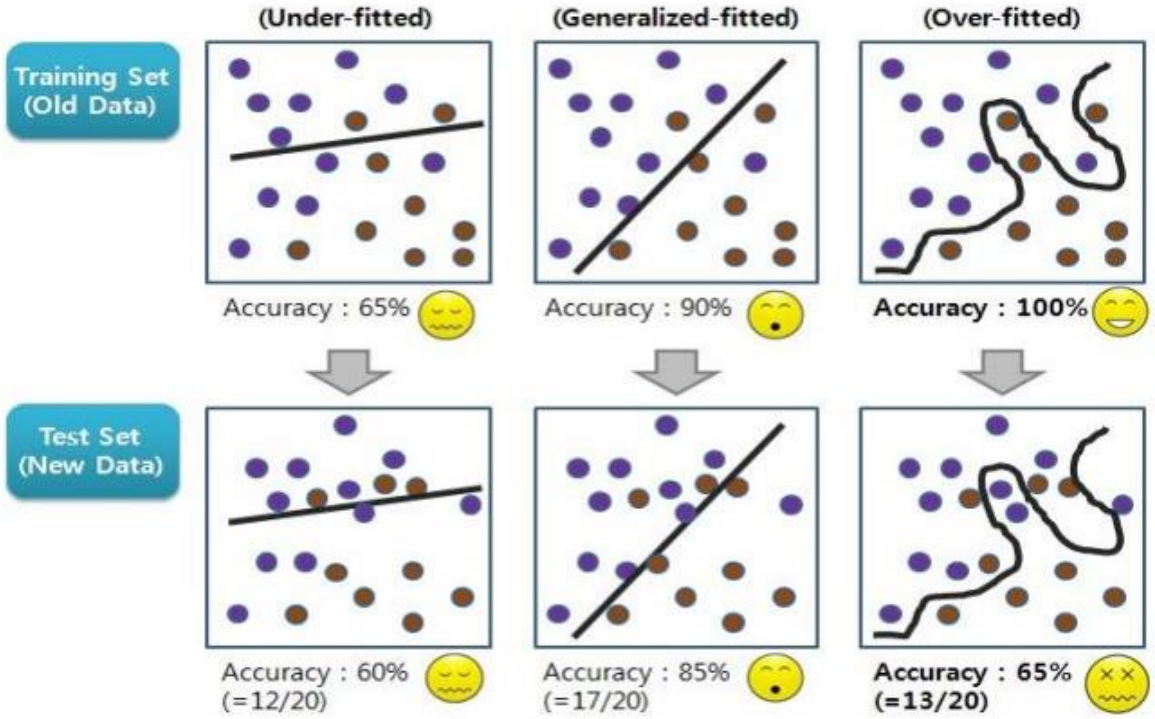
동양미래대학교

장점	단점
• 이해하고 해석하기 쉽다. (**while-box model**) • **데이터 전처리가 상대적으로 적게 필요**함. (데이터 정규화나 특성공학이 상대적으로 덜 필요. 단, 누락값 처리는 해주어야 함.) • **모델 평가가 간단**함 ((O(logN)), N = 트리의 깊이) • **Multi-output 문제도 처리 가능**	• **과적합 발생 가능성**이 높다. ➔ 하이퍼파라미터 튜닝을 잘해야 함) • **데이터에 민감** (데이터가 약간만 달라져도 학습 모델이 민감하게 반응(변경)) → 앙상블 방법으로 해결 • 학습 시, greedy 알고리즘과 같은 Heuristic 알고리즘을 사용하므로 global optimization을 보장할 수 없음. (optimal 결정트리를 찾는 것은 NP-complete 문제) → 앙상블 방법으로 해결 • **데이터 분포가 편중** 시, 성능 저하 → 데이터 전처리 작업 수행

과적합 (Overfitting)

동양미래대학교

학습 데이터에 너무 지나치게 맞추다 보면 일반화 성능이 떨어지는 모델을 얻게 되는 현상을 과적합(Overfitting)이라고 한다.



- **Under Fitting**

  적정 수준의 학습을 하지 못하여 실제 성능이 떨어지는 경우

- **Normal Fitting (Generalized Fitting)**

  적정 수준의 학습으로 실제 적정한 일반화 수준을 나타냄. 기계 학습이 지향하는 수준.

- **Over Fitting**

  학습 데이터에 성능이 좋지만 실제 데이터에 관해 성능이 떨어짐. 특히 조심해야 함.

Scikit-learn 에서의 Decision Tree - Sklearn.tree.DecisionTreeClassifier

동양미래대학교

Parameters:

criterion : *string, optional (default="gini")*

The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.

트리를 구성할 때 사용하는 불순도
[gini, entropy]

splitter : *string, optional (default="best")*

The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

트리를 split할 때의 전략
[best, random]

max_depth : *int or None, optional (default=None)*

The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

최종 생성되는 트리의 Depth의 최대값

min_samples_split : *int, float, optional (default=2)*

The minimum number of samples required to split an internal node:

- If int, then consider `min_samples_split` as the minimum number.
- If float, then `min_samples_split` is a fraction and `ceil(min_samples_split * n_samples)` are the minimum number of samples for each split.

Changed in version 0.18: Added float values for fractions.

노드를 split할 때 필요한 최소 샘플 수

min_samples_leaf : *int, float, optional (default=1)*

The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least `min_samples_leaf` training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression.

- If int, then consider `min_samples_leaf` as the minimum number.
- If float, then `min_samples_leaf` is a fraction and `ceil(min_samples_leaf * n_samples)` are the minimum number of samples for each node.

Changed in version 0.18: Added float values for fractions.

Leaf node에 있는 최소 샘플 수

Scikit-learn 에서의 Decision Tree - Sklearn.tree.DecisionTreeClassifier

동양미래대학교

Parameters:

max_features : *int, float, string or None, optional (default=None)*

The number of features to consider when looking for the best split:

- If int, then consider `max_features` features at each split.
- If float, then `max_features` is a fraction and `int(max_features * n_features)` features are considered at each split.
- If "auto", then `max_features=sqrt(n_features)`.
- If "sqrt", then `max_features=sqrt(n_features)`.
- If "log2", then `max_features=log2(n_features)`.
- If None, then `max_features=n_features`.

Note: the search for a split does not stop until at least one valid partition of the node samples is found, even if it requires to effectively inspect more than `max_features` features.

split할 때 고려하는 feature의 개수

class_weight : *dict, list of dicts, "balanced" or None, default=None*

Weights associated with classes in the form `{class_label: weight}`. If not given, all classes are supposed to have weight one. For multi-output problems, a list of dicts can be provided in the same order as the columns of y.

Note that for multioutput (including multilabel) weights should be defined for each class of every column in its own dict. For example, for four-class multilabel classification weights should be [{0: 1, 1: 1}, {0: 1, 1: 5}, {0: 1, 1: 1}, {0: 1, 1: 1}] instead of [{1:1}, {2:5}, {3:1}, {4:1}].

The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as `n_samples / (n_classes * np.bincount(y))`

For multi-output, the weights of each column of y will be multiplied.

Note that these weights will be multiplied with sample_weight (passed through the fit method) if sample_weight is specified.

Target label의 가중치 부여

- default = None: 모두 같은 가중치 (1)
- balanced: target label 분포에 따라 weight 설정

http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier

DecisionTree 사용 팁

동양미래대학교

1. 차원(feature)이 너무 많거나, 샘플의 크기가 너무 작으면 overfitting되는 경향이 있으므로, 차원이 많은 경우, 차원 축소(PCA, ICA, Feature Selection)를 고려할 필요가 있다.
2. max_depth를 3으로 하고, 우선 트리를 생성해 보고, 점점 max_depth를 점점 늘려가면서 테스트하라.
3. 트리 모델을 도식화해서 보라.
4. 오버피팅 발생 시, 아래 파라미터들을 조절해 보라.
    1. max_depth 낮추기
    2. min_sample_split 높이기 – split 노드의 최소 샘플 개수. 이 개수에 도달하는 노드가 만들어지면 더 이상 트리를 확장하지 않는다.
    3. min_sample_leaf 높이기 – leaf 노드의 최소 샘플 개수. 이 개수에 도달하는 리프 노드가 만들어지면 더 이상 트리를 확장하지 않는다.
5. 트리가 bias되지 않도록 데이터 밸런스를 맞춰라. 예를 들어, 각 클래스별로 동일한 개수의 샘플 데이터 사용.

http://scikit-learn.org/stable/modules/tree.html#tree