

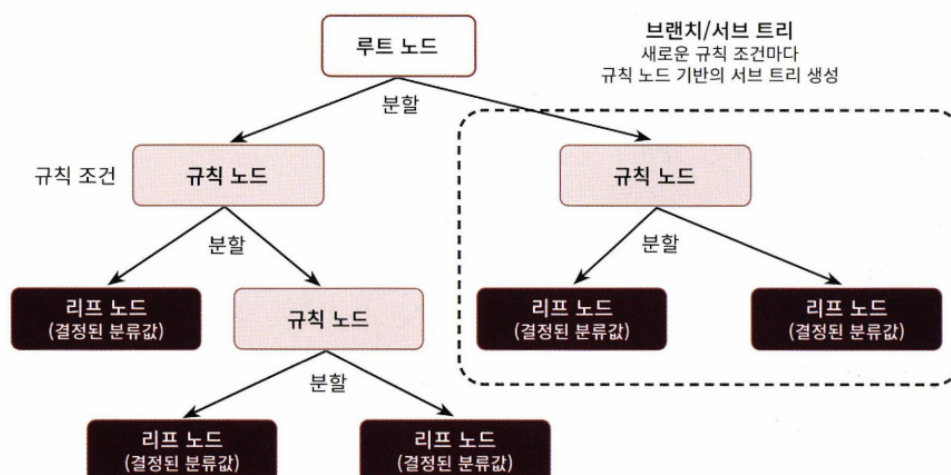
# 4장 분류

## 4.1 분류(Classification)의 개요

- 분류 : 기존 데이터가 어떤 레이블에 속하는지 패턴을 알고리즘으로 인지한 뒤에 새롭게 관측된 데이터 대한 레이블을 판별
- 머신러닝 알고리즘 (분류)
  1. 나이브 베이즈(Naive Bayes) : 베이즈(Bayes) 통계와 생성 모델에 기반한 기반
  2. 로지스틱 회귀(Logistic Regression) : 독립변수와 종속변수의 선형 관계성에 기반
  3. 결정 트리(Decision Tree) : 데이터 균일도에 따른 규칙 기반
  4. 서포트 벡터 머신(Support Vector Machine) : 개별 클래스 간의 최대 분류 마진을 효과적으로 찾는 기법
  5. 최소 근접 알고리즘(Nearest Neighbor) : 근접 거리를 기준으로 하는 기법
  6. 신경망(Neural Network) : 심층 연결 기반
  7. 앙상블(Ensemble) : 서로 다른(또는 같은) 머신러닝 알고리즘을 결합한 기법

## 4.2 결정 트리

- 결정 트리 : 데이터에 있는 규칙을 학습을 통해 자동으로 찾아내 트리(Tree) 기반의 분류 규칙을 만드는 것(if/else 기반)



- 위의 그림에서 보듯이 트리의 깊이(depth)가 깊어질수록 결정 트리의 예측 성능이 저하될 가능성이 높다.

- 결정 트리의 장단점

결정 트리 장점	결정 트리 단점
<ul style="list-style-type: none"> <li>• 쉽다. 직관적이다</li> <li>• 피처의 스케일링이나 정규화 등의 사전 가공 영향도가 크지 않음.</li> </ul>	<ul style="list-style-type: none"> <li>• 과적합으로 알고리즘 성능이 떨어진다. 이를 극복하기 위해 트리의 크기를 사전에 제한하는 튜닝 필요.</li> </ul>

- 결정 트리 주요 함수

DecisionTreeClassifier 객체의 `feature_importances` : ndarray 형태로 값을 반환하며 피처 순서대로 값이 할당된다. → 즉 피처별 결정 트리 알고리즘에서의 중요도 파악 가능

`make_classification` : 분류를 위한 테스트용 데이터를 쉽게 만드는 API 함수

→ 책에서는 이를 이용해 이상치 탐지를 시각화하는데 사용

- 노드 결정 기준 : (정보이득, 지니계수 이용)

- 결정 트리 실습 : 코드를 통해 참고

- idea : 피처명을 txt파일로 불러오는데 이 과정에서 중복된 데이터들이 많다. 따라서 이를 전처리하는 과정을 유심히 살펴보기로 한다.

## 추가 공부 : 엔트로피와 정보 이득

- 간단히 말하자면 각 데이터 분할에 따른 불순도(종교 나뭇잎)를 지니계수와, 엔트로피에 따라 계산하고 이를 정보이득 즉 `information gain`으로 종합하여 어떤 게 더 나은 선택 인지 판단해 가는 작업으로 결정트리가 정해지는것!
- 그렇다면, 그 전의 정보 이론은 무엇이고, 엔트로피란 무엇인지 먼저 살펴보자!

## 분류

엔트로피(entropy)와 정보이론

정보이론에서는 정보량을 나타내기 위해 엔트로피라는 단위를 사용합니다. 물리에서 쓰이는 엔트로피처럼 정보이론의 엔트로피도 불확실한 정보를 숫자로 정량화하려는 노력이자 하나의 도구입니다.

엔트로피는 간단히 말하자면 확률의 역수에 로그를 취한 값입니다. 어떤 사건  $X$ 가 일어날 확률을  $p(x)$ , 엔트로피를  $h(x)$ 라고 할 때 엔트로피는 아래의 식으로 정의할 수 있습니다.

$$h(x) = \log \frac{1}{p(x)} = -\log p(x)$$

확률의 역수를 취해주는 이유는 확률이 높은 사건일수록 정보량(놀라움)이 적다고 판단하기 때문입니다. 내일 비가 올 확률이 1%일 때 비가 오지 않을 확률은 99%일 것입니다. 이때 각 사건의 정보량은 다음과 같습니다.

$$\begin{aligned} h(\text{비}) &= -\log 0.01 = 4.605 \\ h(\text{비가 오지 않음}) &= -\log 0.99 = 0.010 \end{aligned}$$

비가 오지 않는 경우는 비가 오는 경우보다 460배 정도 더 놀라운 사건이 됩니다.

엔트로피의 기댓값은 각 엔트로피에 확률을 곱해준 값입니다. 통계에서 기댓값의 정의는 각 사건이 벌어졌을 때의 이득과 그 사건이 벌어질 확률을 곱한 값이기 때문입니다. 역시 수식으로 나타내면 아래와 같습니다.

$$E(X) = -p(x) \log p(x)$$

시작합니다! Artificial Intelligence

- summary : 정보이론과 엔트로피의 소개 및 계산과정 소개
- #참고 영상 - 새년의 정보이론
  - [https://www.youtube.com/watch?v=d3iyDP3\\_AjU](https://www.youtube.com/watch?v=d3iyDP3_AjU)

## 머신러닝

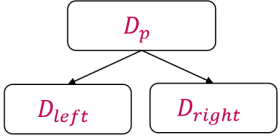
결정 알고리즘을 사용하면 트리의 루트(root)에서 시작해 정보 이득(Information Gain, IG)이 최대가 되는 특성으로 데이터를 나눕니다. 정보 이득에 대해서는 뒤에 이어서 설명을 드리겠습니다.

결과적으로 IG가 최대가 되는 특성으로 데이터를 나누는 반복작업을 통해 리프 노드(leaf node)가 순수해질 때까지(더 이상 나눌 수 없을 때까지) 모든 자식 노드에서 이 분할 작업을 반복합니다. 실제로 이렇게 하면 노드가 많은 깊은 트리가 만들어지고 과대적합(과적합)될 가능성이 높습니다. 일반적으로 트리의 최대 깊이를 제한하여 트리를 가지치기(pruning) 합니다.

그렇다면 정보 이득이란 무엇일까요?

정보 이득(IG) 최대화 : 자원을 최대한 활용

가장 정보가 풍부한 특성으로 노드를 나누기 위해 트리 알고리즘으로 최적화할 목적 함수를 정의합니다. 이 목적 함수는 각 분할에서 정보 이득을 최대화 합니다. 정보 이득은 다음과 같이 정의할 수 있습니다.


$$IG(D_p, f) = I(D_p) - \sum_{j=1}^m \frac{N_j}{N_p} I(D_j)$$
$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

여기서  $f$ 는 분할에 사용할 특성입니다.  $D_p$ 와  $D_j$ 는 부모와  $j$ 번째 자식 노드의 데이터셋입니다.  $I$ 는 불순도(impurity) 지표입니다.  $N_p$ 는 부모 노드에 있는 전체 데이터의 개수이며,  $N_j$ 는  $j$ 번째 자식 노드에 있는 데이터의 개수입니다.

여기서 볼 수 있듯이 정보 이득은 단순히 부모 노드의 불순도와 자식 노드의 불순도 합차이입니다. 위의 수식에 의하면 자식 노드의 불순도가 낮을수록 정보 이득은 커지게 됩니다. 구현을 간단하게 하고 탐색 공간을 줄이기 위해 대부분의 라이브러리는 이진 결정트리를 사용합니다. 즉, 부모 노드는 두 개의 자식 노드  $D_{left}$ 와  $D_{right}$ 로 나뉘집니다.

- summary : 정보 이득(information gain)이 목적함수로서 작용을 하고 이를 최대화 하는 방식으로 계산된다. 또한 의미적으로 살펴보면 정보 이득은 부모 노드와 자식 노드간

의 불순도 합의 차이이다. 즉, 이를 최대화 한다는 목적함수는 자식 노드의 불순도를 낮추는 방향으로 진행됨을 알 수 있다.

## 머신러닝

이진 결정 트리에 널리 사용되는 세 개의 불순도 지표 또는 분할 조건은 지니 불순도(Gini impurity,  $I_G$ ), 엔트로피(entropy,  $I_H$ ), 분류 오차(classification error,  $I_E$ )입니다. 데이터가 있는 모든 클래스 ( $p(i|t) \neq 0$ )에 대한 엔트로피 정의는 다음과 같습니다.

$$I_H(t) = - \sum_{i=1}^c p(i|t) \log_2 p(i|t)$$

여기서  $p(i|t)$ 는 특정 노드  $t$ 에서 클래스  $i$ 에 속한 샘플 비율입니다. 한 노드의 모든 샘플이 같은 클래스이면 엔트로피는 0이 됩니다. 반대로 한 노드의 클래스 분포가 균등하면 엔트로피는 최대가 됩니다.

예를 들어 이진 클래스일 경우  $p(i = 1|t) = 1$  또는  $p(i = 0|t) = 0$ 이면 엔트로피는 0입니다. (※ 한 노드의 모든 샘플이 같은 클래스일 확률 0% or 100%) 클래스가  $p(i = 1|t) = 0.5$ 와  $p(i = 0|t) = 0.5$  처럼 균등하게 분포되어 있으면 엔트로피는 1이 됩니다.

▶ 엔트로피 조건을 트리의 상호 의존 정보를 최대화 하는 것으로 이해할 수 있습니다.

(※  $\log_2(0.5)$ 는 -1이므로 두 노드가 균등하게 분포되어 있으면  $I_H(t) = -(0.5 * (-1) + 0.5 * (-1)) = 1$ )

```
>>> np.log2(0.5)
-1.0
```

## 시작합니다! Artificial Intelligence

- summary : 불순도 지표에는 여러가지가 있는데 이 중 엔트로피를 소개하는 슬라이드, 엔트로피를 계산한 후에 이를 최대화 하는 방향으로 진행(지표가 이 슬라이드등에서는 엔트로피이고 다른 지표로 지니 계수 등이 있음 → 아래 슬라이드 참고)

## 머신러닝

자연스럽게 지니 불순도는 잘못 분류될 확률을 최소화 하기 위한 기준으로 이해할 수 있습니다.

$$I_G(t) = \sum_{i=1}^c p(i|t)(1 - p(i|t)) = 1 - \sum_{i=1}^c p(i|t)^2$$

엔트로피와 비슷하게 지니 불순도는 클래스가 완벽하게 섞여 있을 때(0.5) 최대가 됩니다. 예를 들어 이진 클래스 환경( $c = 2$ )에서는 다음과 같습니다.

$$I_G(t) = 1 - \sum_{i=1}^c 0.5^2 = 0.5$$

실제로 지니 불순도와 엔트로피 모두 매우 비슷한 결과가 나옵니다. 보통 불순도 조건을 바꾸어 트리를 평가하는 것보다 가지치기 수준을 바꿔가며 튜닝하는 것이 훨씬 낫습니다.

또 다른 불순도 지표는 분류 오차입니다. 분류 오차 불순도 지표도 마찬가지로 두 클래스가 같은 비율일 때 최대(0.5)가 되고 한 클래스의 비율이 커질수록 줄어듭니다.

$$I_E = 1 - \max\{p(i|t)\}$$

가지치기에는 좋은 기준이지만 결정 트리를 구성하는 데는 권장되지 않습니다. 노드의 클래스 확률 변화에 덜 민감하기 때문입니다. 다음 그림을 통해 두 개의 분할 시나리오를 알아보겠습니다.

## 시작합니다! Artificial Intelligence

- summary : 불순도 지표의 다른 기준인 지니불순도(지니계수)를 소개, 추가적으로 간단히 두 자식 노드의 차이를 나타내는 불순도 지표인 분류 오차를 소개

## 머신러닝



부모 노드에서 데이터셋  $D_p$ 로 시작합니다. 이 데이터셋은 클래스 1이 40개의 데이터, 클래스 2가 40개의 데이터로 이루어져 있습니다. 이를 두 개의 데이터셋  $D_{left}$ ,  $D_{right}$ 로 나눕니다.

$$I_E = 1 - \max\{p(i|t)\}$$

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

분류 오차를 분할 기준으로 사용했을 때 정보 이득은 시나리오 A, B가 동일합니다. ( $IG_E = 0.25$ )

$$I_E(D_p) = 1 - 0.5 = 0.5$$

$$A : I_E(D_{left}) = 1 - \frac{3}{4} = 0.25$$

$$A : I_E(D_{right}) = 1 - \frac{3}{4} = 0.25$$

$$A : IG_E = 0.5 - \frac{4}{8} 0.25 - \frac{4}{8} 0.25 = 0.25$$

$$B : I_E(D_{left}) = 1 - \frac{4}{6} = \frac{1}{3}$$

$$B : I_E(D_{right}) = 1 - 1 = 0$$

$$B : IG_E = 0.5 - \frac{6}{8} \frac{1}{3} - 0 = 0.25$$

- summary : 불순도 지표들이 계산되는 과정을 소개( 분류 오차를 기준)

## 머신러닝

$$I_G(t) = 1 - \sum_{i=1}^c 0.5^2 = 0.5$$

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N_p} I(D_{left}) - \frac{N_{right}}{N_p} I(D_{right})$$

지니 불순도는 시나리오 A( $IG_G = 0.125$ ) 보다 시나리오 B( $IG_G = 0.16$ )가 더 순수하기 때문에 값이 높습니다.

$$I_G(D_p) = 1 - (0.5^2 + 0.5^2) = 0.5$$

$$A : I_G(D_{left}) = 1 - \left( \left( \frac{3}{4} \right)^2 + \left( \frac{1}{4} \right)^2 \right) = \frac{3}{8} = 0.375$$

$$A : I_G(D_{right}) = 1 - \left( \left( \frac{1}{4} \right)^2 + \left( \frac{3}{4} \right)^2 \right) = \frac{3}{8} = 0.375$$

$$A : IG_G = 0.5 - \frac{4}{8} 0.375 - \frac{4}{8} 0.375 = 0.125$$

$$B : I_G(D_{left}) = 1 - \left( \left( \frac{2}{6} \right)^2 + \left( \frac{4}{6} \right)^2 \right) = \frac{4}{9} = 0.\bar{4}$$

$$B : I_G(D_{right}) = 1 - (1^2 + 0^2) = 0$$

$$B : IG_G = 0.5 - \frac{6}{8} 0.\bar{4} - 0 = 0.1\bar{6}$$

시작합니다! Artificial Intelligence

- summary : 불순도 지표들이 계산되는 과정을 소개( 지니 불순도를 기준)

비슷하게 엔트로피 기준도 시나리오 A( $IG_H = 0.19$ ) 보다 시나리오 B( $IG_H = 0.31$ )를 선호합니다.

$$I_H(D_p) = -(0.5 \log_2(0.5) + 0.5 \log_2(0.5)) = 1$$

$$A : I_H(D_{left}) = -\left(\frac{3}{4} \log_2\left(\frac{3}{4}\right) + \frac{1}{4} \log_2\left(\frac{1}{4}\right)\right) = 0.81$$

$$B : I_H(D_{left}) = -\left(\frac{2}{6} \log_2\left(\frac{2}{6}\right) + \frac{4}{6} \log_2\left(\frac{4}{6}\right)\right) = 0.92$$

$$A : I_H(D_{right}) = -\left(\frac{1}{4} \log_2\left(\frac{1}{4}\right) + \frac{3}{4} \log_2\left(\frac{3}{4}\right)\right) = 0.81$$

$$B : I_H(D_{right}) = 0$$

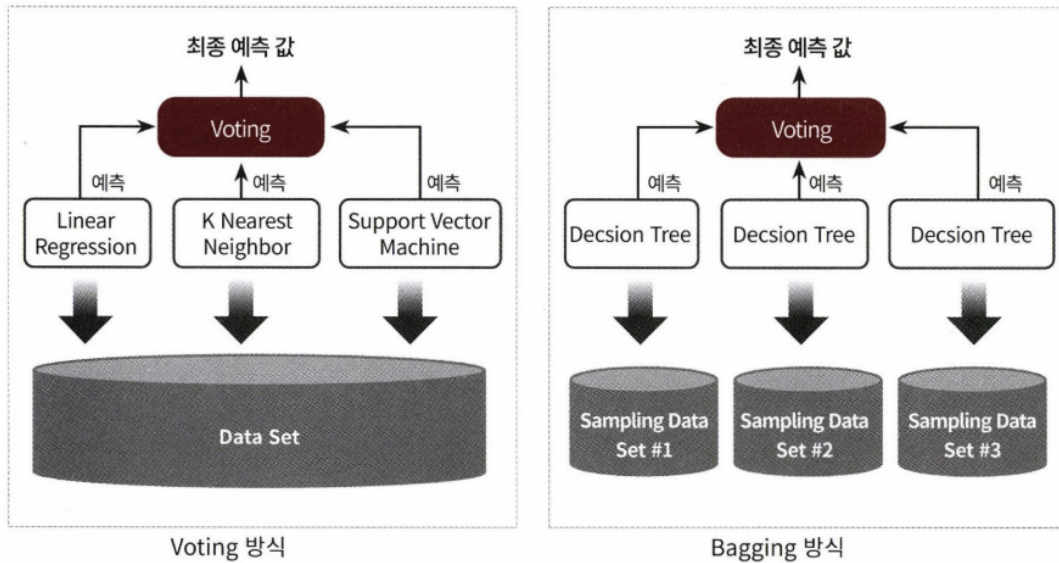
$$A : IG_H = 1 - \frac{4}{8} \cdot 0.81 - \frac{4}{8} \cdot 0.81 = 0.19$$

$$B : IG_H = 1 - \frac{6}{8} \cdot 0.92 - 0 = 0.31$$

- summary : 불순도 지표들이 계산되는 과정을 소개( 엔트로피 오차를 기준)
- 결정 트리 뿐만 아니라 엔트로피는 정말 다양하게 쓰이기 때문에 이번 기회에 확실히 알아가시면 좋을것 같아요

### 4.3 앙상블 학습

- 앙상블 학습의 유형 : 보팅(Voting), 배깅(Bagging), 부스팅(Boosting), 스택킹(Stacking)
- 보팅(voting) : 일반적으로 서로 다른 알고리즘을 가진 분류기를 결합하는것
- 배깅(Bagging) : 같은 유형의 알고리즘 기반이지만, 데이터 샘플링을 서로 다르게 가져가면서 학습을 수행해 보팅을 수행하는 것



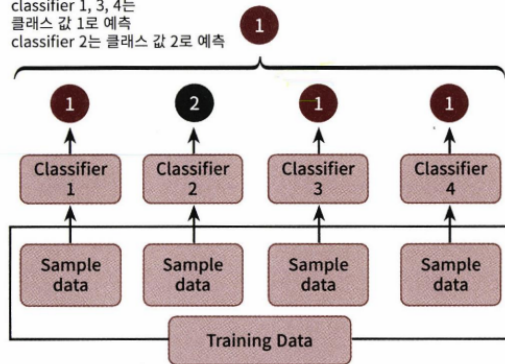
- 위의 오른쪽 그림처럼 개별 분류기에 할당된 학습데이터는 원본 학습 데이터를 샘플링 하여 추출된다, 이렇게 개별 Classifier에게 데이터를 샘플링해서 추출하는 방식을 **부트스트래핑(Bootstrapping)** 분할 방식이라고 부른다.
- 부스팅 : 여러 개의 분류기가 순차적으로 학습을 수행하되 앞에서 학습한 분류기가 예측이 틀린 데이터에 대해서는 올바르게 예측할 수 있도록 다음 분류기에서는 가중치 (weight)를 부여하면서 학습과 예측을 진행하는 것,
- ex) 그라디언트 부스트, XGBoost(eXtra Gradient Boost), LightGBM(Light Gradient Boost)
- 스태킹 : 여러 가지 다른 모델의 예측 결과값을 다시 학습데이터로 만들어서 다른 모델 (메타 모델로)로 재학습시켜 결과를 예측하는 방법 (즉 앙상블의 앙상블)

보팅 유형 - 하드 보팅(Hard Voting), 소프트 보팅(Soft Voting)

- 하드 보팅 : 다수의 분류기가 결정한 예측값을 최종 보팅 결과값으로 선정
- 소프트 보팅 : 분류기들의 레이블 값 결정 확률을 모두 더하고 이를 평균해서 이들 중 확률이 가장 높은 레이블 값을 최종 보팅 결과값으로 선정

Hard Voting은 다수의 classifier 간 다수결로 최종 class 결정

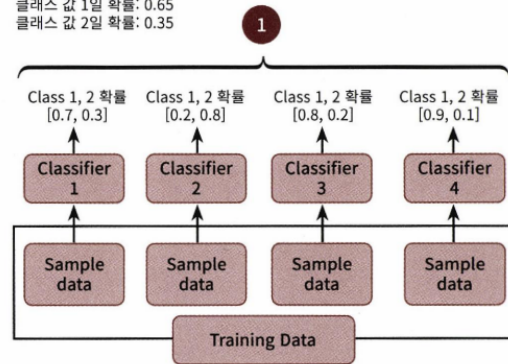
클래스 값 1로 예측  
 classifier 1, 3, 4는  
 클래스 값 1로 예측  
 classifier 2는 클래스 값 2로 예측



< 하드 보팅 >

Soft Voting은 다수의 classifier 들의 class 확률을 평균하여 결정

클래스 값 1로 예측  
 클래스 값 1일 확률: 0.65  
 클래스 값 2일 확률: 0.35



< 소프트 보팅 >