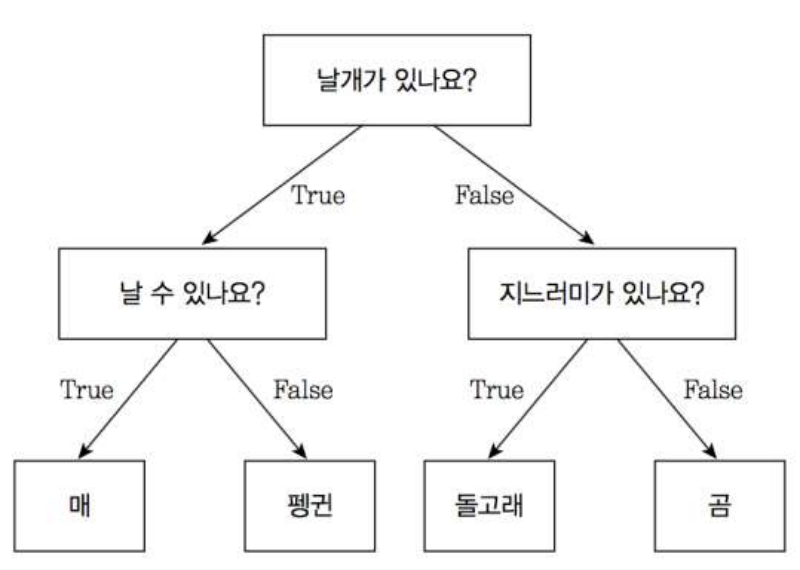
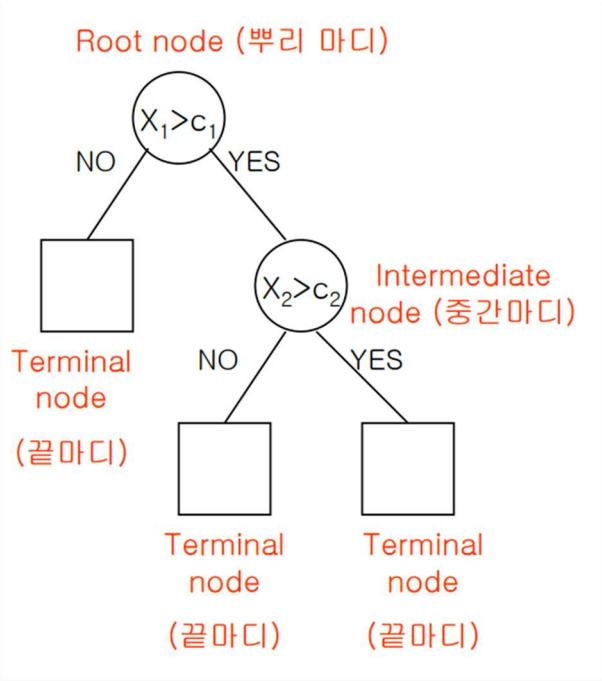
의사 결정 트리(Decision Tree)

결정 트리(Decision Tree, 의사결정트리, 의사결정나무라고도 함)는 분류(Classification)와 회귀(Regression) 모두 가능한 지도 학습 모델 중 하나입니다. 결정 트리는 스무고개 하듯이 예/아니오 질문을 이어가며 학습합니다. 매, 펭귄, 돌고래, 곰을 구분한다고 생각해봅시다. 매와 펭귄은 날개를 있고, 돌고래와 곰은 날개가 없습니다. '날개가 있나요?'라는 질문을 통해 매, 펭귄 / 돌고래, 곰을 나눌 수 있습니다. 매와 펭귄은 '날 수 있나요?'라는 질문으로 나눌 수 있고, 돌고래와 곰은 '지느러미가 있나요?'라는 질문으로 나눌 수 있습니다. 아래는 결정 트리를 도식화한 것입니다.



출처: 텐서 플로우 블로그

이렇게 특정 기준(질문)에 따라 데이터를 구분하는 모델을 결정 트리 모델이라고 합니다. 한번의 분기 때마다 변수 영역을 두 개로 구분합니다. 결정 트리에서 질문이나 정답을 담은 네모 상자를 노드(Node)라고 합니다. 맨 처음 분류 기준 (즉, 첫 질문)을 Root Node 라고 하고, 맨 마지막 노드를 Terminal Node 혹은 Leaf Node 라고 합니다.

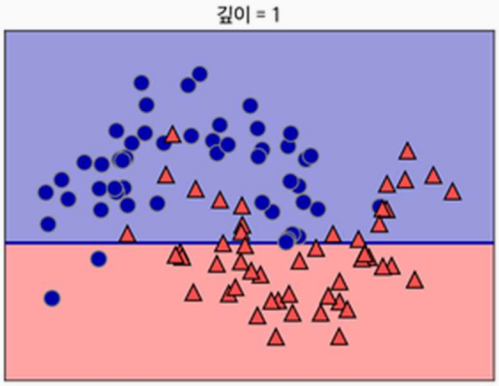


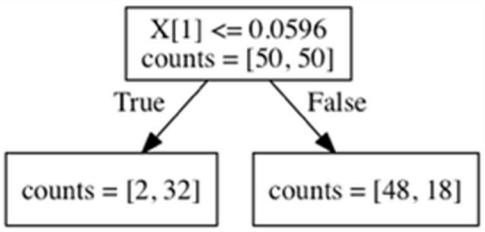
출처: ratsgo's blog

전체적인 모양이 나무를 뒤집어 놓은 것과 같아서 이름이 Decision Tree 입니다.

프로세스

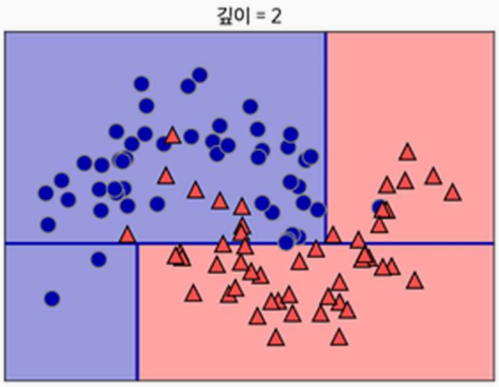
결정 트리 알고리즘의 프로세스를 간단히 알아보겠습니다.

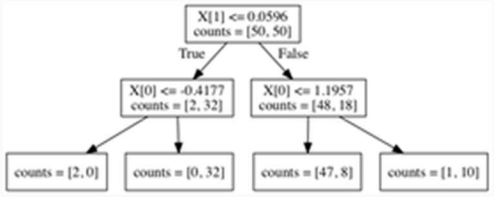




출처: 텐서플로우 블로그

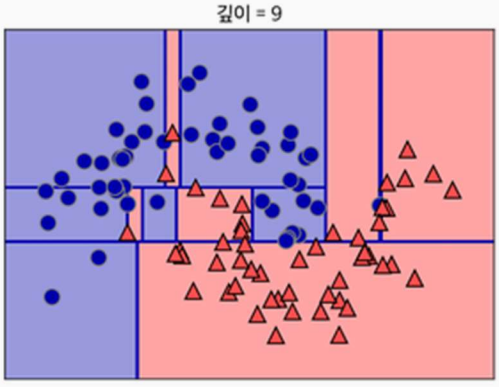
먼저 위와 같이 데이터를 가장 잘 구분할 수 있는 질문을 기준으로 나눕니다.





출처: 텐서플로우 블로그

나뉜 각 범주에서 또 다시 데이터를 가장 잘 구분할 수 있는 질문을 기준으로 나눕니다. 이를 지나치게 많이 하면 아래와 같이 오버피팅이 됩니다. 결정 트리에 아무 파라미터를 주지 않고 모델링하면 오버피팅이 됩니다.



출처: 텐서플로우 블로그

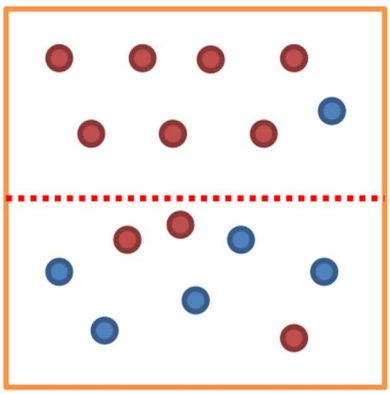
가지치기(Pruning)

오버피팅을 막기 위한 전략으로 가지치기(Pruning)라는 기법이 있습니다. 트리에 가지가 너무 많다면 오버피팅이라 볼 수 있습니다. 가지치기란 나무의 가지를 치는 작업을

말합니다. 즉, 최대 깊이나 터미널 노드의 최대 개수, 혹은 한 노드가 분할하기 위한 최소 데이터 수를 제한하는 것입니다. min_sample_split 파라미터를 조정하여 한 노드에 들어있는 최소 데이터 수를 정해줄 수 있습니다. min_sample_split = 10 이면 한 노드에 10 개의 데이터가 있다면 그 노드는 더 이상 분기를 하지 않습니다. 또한, max_depth 를 통해서 최대 깊이를 지정해줄 수도 있습니다. max_depth = 4 이면, 깊이가 4 보다 크게 가지를 치지 않습니다. 가지치기는 사전 가지치기와 사후 가지치기가 있지만 sklearn 에서는 사전 가지치기만 지원합니다.

알고리즘: 엔트로피(Entropy), 불순도(Impurity)

불순도(Impurity)란 해당 범주 안에 서로 다른 데이터가 얼마나 섞여 있는지를 뜻합니다. 아래 그림에서 위쪽 범주는 불순도가 낮고, 아래쪽 범주는 불순도가 높습니다. 바꾸어 말하면 위쪽 범주는 순도(Purity)가 높고, 아래쪽 범주는 순도가 낮습니다. 위쪽 범주는 다 빨간점인데 하나만 파란점이므로 불순도가 낮다고 할 수 있습니다. 반면 아래쪽 범주는 5 개는 파란점, 3 개는 빨간점으로 서로 다른 데이터가 많이 섞여 있어 불순도가 높습니다.



한 범주에 하나의 데이터만 있다면 불순도가 최소(혹은 순도가 최대)이고, 한 범주 안에 서로 다른 두 데이터가 정확히 반반 있다면 불순도가 최대(혹은 순도가 최소)입니다. 결정 트리는 불순도를 최소화(혹은 순도를 최대화)하는 방향으로 학습을 진행합니다.

엔트로피(Entropy)는 불순도(Impurity)를 수치적으로 나타낸 척도입니다. 엔트로피가 높다는 것은 불순도가 높다는 뜻이고, 엔트로피가 낮다는 것은 불순도가 낮다는 뜻입니다. 엔트로피가 1 이면 불순도가 최대입니다. 즉, 한 범주 안에 서로 다른 데이터가 정확히 반반 있다는 뜻입니다. 엔트로피가 0 이면 불순도는 최소입니다. 한 범주 안에 하나의 데이터만 있다는 뜻입니다. 엔트로피를 구하는 공식은 아래와 같습니다.

$$\text{Entropy} = -\sum_i (p_i) \log_2 (p_i)$$ 엔트로피 공식

(Pi = 한 영역 안에 존재하는 데이터 가운데 범주 i 에 속하는 데이터의 비율)

엔트로피 예제

공식을 활용하여 엔트로피를 구하는 예제를 살펴보겠습니다. 아래는 경사, 표면, 속도 제한을 기준으로 속도가 느린지 빠른지 분류해놓은 표입니다.

경사	표면	속도 제한	속도
steep	bumpy	yes	slow
steep	smooth	yes	slow
flat	bumpy	no	fast
steep	smooth	no	fast

첫줄을 보면 경사가 가파르고(steep), 표면이 울퉁불퉁하고(bumpy), 속도 제한이 있다면(yes) 속도가 느리다(slow)라고 분류했습니다. X variables 가 경사, 표면, 속도

제한이고, Y variable 이 속도입니다. 이때 엔트로피를 기반으로 결정 트리 모델링 해보겠습니다.

속도 라벨에는 slow, slow, fast, fast 로 총 4 개의 examples 가 있습니다. Pi 는 한 영역 안에 존재하는 데이터 가운데 범주 i 에 속하는 데이터의 비율이라고 했습니다. i 를 slow 라고 했을 때, P_slow = 0.5 입니다. slow 라벨 갯수 = 2 개, 전체 examples 수 = 4 개이기 때문에 P_slow = 2/4 = 0.5 입니다. 마찬가지로, P_fast 도 0.5 입니다. fast 라벨 갯수가 2 개이기 때문에 2/4 로 0.5 입니다. 즉 P_slow = 0.5, P_fast = 0.5 입니다.

그렇다면 현재 범주 전체의 엔트로피는 얼마일까요? 바로 1 입니다. 서로 다른 데이터가 정확히 반반 있기 때문입니다. 위에서 봤던 엔트로피 공식에 그래도 대입을 해 엔트로피를 구해보겠습니다.

$$\text{Entropy} = -\sum_i (p_i) \log_2 (p_i)$$ 엔트로피 공식

Entropy = - P_slow * $\log_2$(P_slow) - P_fast * $\log_2$(P_fast) = - 0.5 * $\log_2$(0.5) - 0.5 * $\log_2$(0.5) = 1

공식에 의해서도 1 이라고 구할 수 있고, 데이터가 정확히 반반 (slow 2 개, fast 2 개)이므로 1 이라고 할 수 있습니다.

정보 획득 (Information gain)

엔트로피가 1 인 상태에서 0.7 인 상태로 바뀌었다면 정보 획득(information gain)은 0.3 입니다. 분기 이전의 엔트로피에서 분기 이후의 엔트로피를 뺀 수치를 정보 획득이라고 합니다. 정보 획득은 아래와 같이 공식화할 수 있습니다.

Information gain = entropy(parent) - [weighted average] entropy(children)

entropy(parent)는 분기 이전 엔트로피이고, entropy(children)은 분기 이후 엔트로피입니다. 이때, [weighted average] entropy(children)는 entropy(childeren)의 가중 평균을 뜻합니다. 분기 이후 엔트로피에 대해 가중 평균을 하는 이유는 분기를 하면 범주가 2 개 이상으로 쪼개지기 때문입니다. 범주가 하나라면 위 엔트로피 공식으로 바로 엔트로피를 구할 수 있습니다. 하지만 범주가 2 개 이상일 경우 가중 평균을 활용하여 분기 이후 엔트로피를 구하는 것입니다.

결정 트리 알고리즘은 정보 획득을 최대화하는 방향으로 학습이 진행됩니다. 어느 feature 의 어느 분기점에서 정보 획득이 최대화되는지 판단을 해서 분기가 진행됩니다.

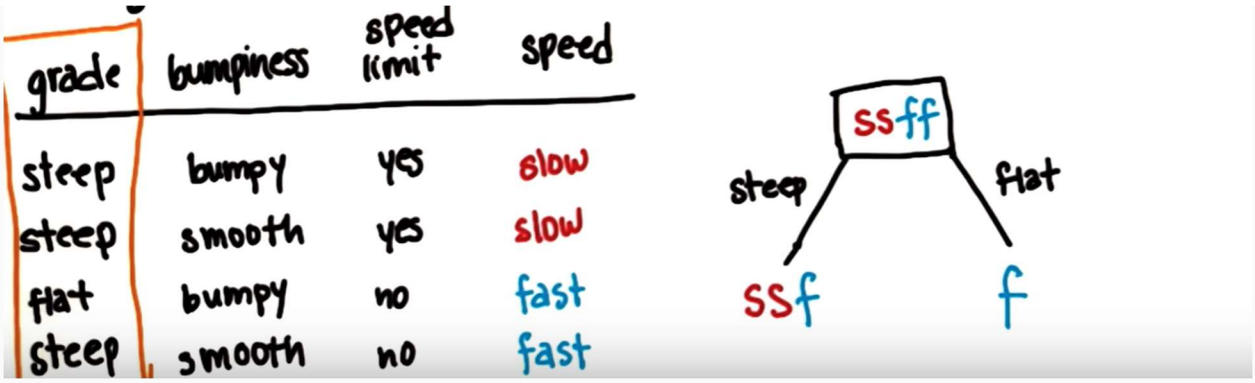
위에서 예로 든 속도 문제를 다시 보겠습니다. 맨 처음 전체 엔트로피 = 1 이라고 했습니다. 즉, entropy of parent = 1 입니다.

entropy(parent) = 1

경사 기준 분기

우선, 경사(grade)를 기준으로 첫 분기를 해보겠습니다. 전체 데이터 중 steep 는 3 개, flat 는 1 개 있습니다. steep 와 flat 을 기준으로 분기를 해주면 결과는 아래와 같습니다. steep 에 해당하는 데이터는 총 3 개이며, 이때의 속도는 slow, slow, fast 입니다. 반면 flat 에 해당하는 데이터는 1 개이며, 이때의 속도는 fast 입니다. flat 에 해당하는 노드의 엔트로피는 얼마일까요? (아래 그림에서 오른쪽 노드) 한 노드에 fast 라는 하나의 데이터만 존재하므로 엔트로피는 0 입니다.

따라서, entropy(flat) = 0 입니다.



출처: Udacity

이제 entropy(steep). 즉, steep 로 분기했을 때의 엔트로피를 구해보겠습니다. 이는 위 그림의 왼쪽 노드에 해당합니다. slow 가 2 개, fast 가 1 개이므로 P_slow = 2/3, P_fast = 1/3 입니다. 따라서 엔트로피 공식에 의해

entropy(steep) = - P_slow * log2(P_slow) - P_fast * log2(P_fast) = - (2/3) * log2(2/3) - (1/3) * log2(1/3) = 0.9184 입니다.

entropy(flat) = 0 이고, entropy(steep) = 0.9184 입니다. 이제 분기 이후 노드에 대한 가중 평균을 구해보겠습니다.

[weighted average] entropy(children) = weighted average of steep * entropy(steep) + weighted average of flat * entropy(flat) = 3/4 * (0.9184) + 1/4 * (0) = 0.6888

(weighted average of steep = 3/4 인 이유는 4 개의 데이터 중 steep 에 해당하는 데이터는 3 개이기 때문입니다. 마찬가지로 weighted average of flat = 1/4 인 이유는 4 개의 데이터 중 flat 에 해당하는 데이터는 1 개이기 때문입니다.)

따라서, 경사(grade)를 기준으로 분기한 후의 엔트로피는 0.6888 입니다. 이제 정보 획득 공식을 통해 정보 획득량을 구해보겠습니다.

information gain = entropy(parent) - [weighted average] entropy(children) = 1 - 0.6888 = 0.3112

경사 feature 를 기준으로 분기를 했을 때는 0.3112 만큼의 정보 획득(information gain)이 있다는 뜻입니다.

표면 기준 분기

표면(bumpiness)을 기준으로 분기했을 때는 bumpy 에는 slow, fast, smooth 에도 slow, fast 가 있습니다. 하나의 범주에 서로 다른 데이터가 정확히 반반 있습니다. 이럴 때 엔트로피는 1 입니다. 공식으로 계산을 해보겠습니다.

entropy(bumpy) = - P_slow * log2(P_slow) - P_fast * log2(P_fast) = - (1/2) * log2(1/2) - (1/2) * log2(1/2) = 1

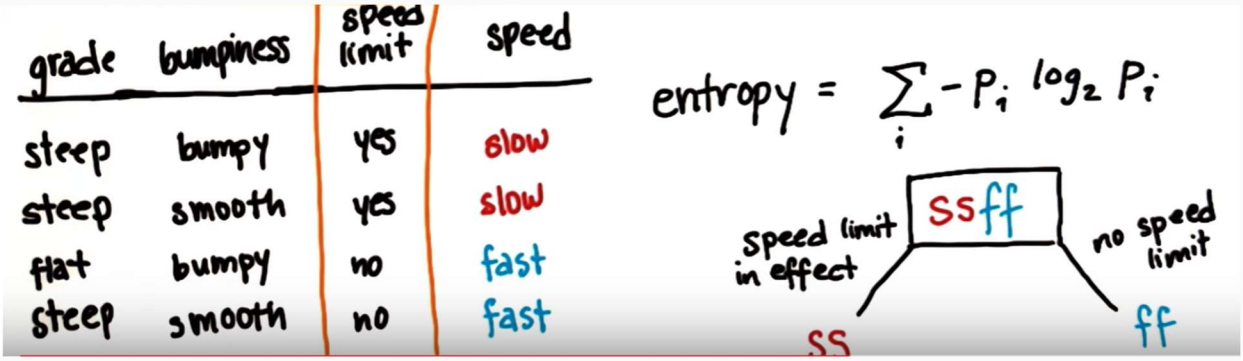
entropy(smooth) = - P_slow * log2(P_slow) - P_fast * log2(P_fast) = - (1/2) * log2(1/2) - (1/2) * log2(1/2) = 1

입니다. 따라서,

information gain = entropy(parent) - [weighted average] entropy(children) = 1 - (2/4) * 1 - (2/4) * 1 = 0 입니다.

표면을 기준으로 분기했을 때는 정보 획득이 전혀 없다는 뜻입니다.

속도제한 기준 분기



출처: Udacity

마찬가지로 계산을 해보면

entropy(yes) = - P_slow * log2(P_slow) - P_fast * log2(P_fast) = - (1) * log2(1) - (0) * log2(0) = 0

entropy(no) = - P_slow * log2(P_slow) - P_fast * log2(P_fast) = - (0) * log2(0) - (1) * log2(1) = 0

입니다. 따라서,

information gain = 1 - (2/4) * 0 - (2/4) * 0 = 1 입니다.

경사, 표면, 속도제한 기준으로 분기 했을 때 정보 획득은 각각 0.3112, 0, 1 입니다. 결정트리는 정보 획득이 가장 많은 방향으로 학습이 진행된다고 했습니다. 따라서 첫 분기점을 속도제한 기준으로 잡습니다. 이런식으로 max_depth 나 min_sample_split 으로 설정한 범위까지 분기를 하게 됩니다. 이것이 바로 결정트리의 전체적인 알고리즘입니다.

Reference: 이기창, 2019, ttps://bkshin.tistory.com/