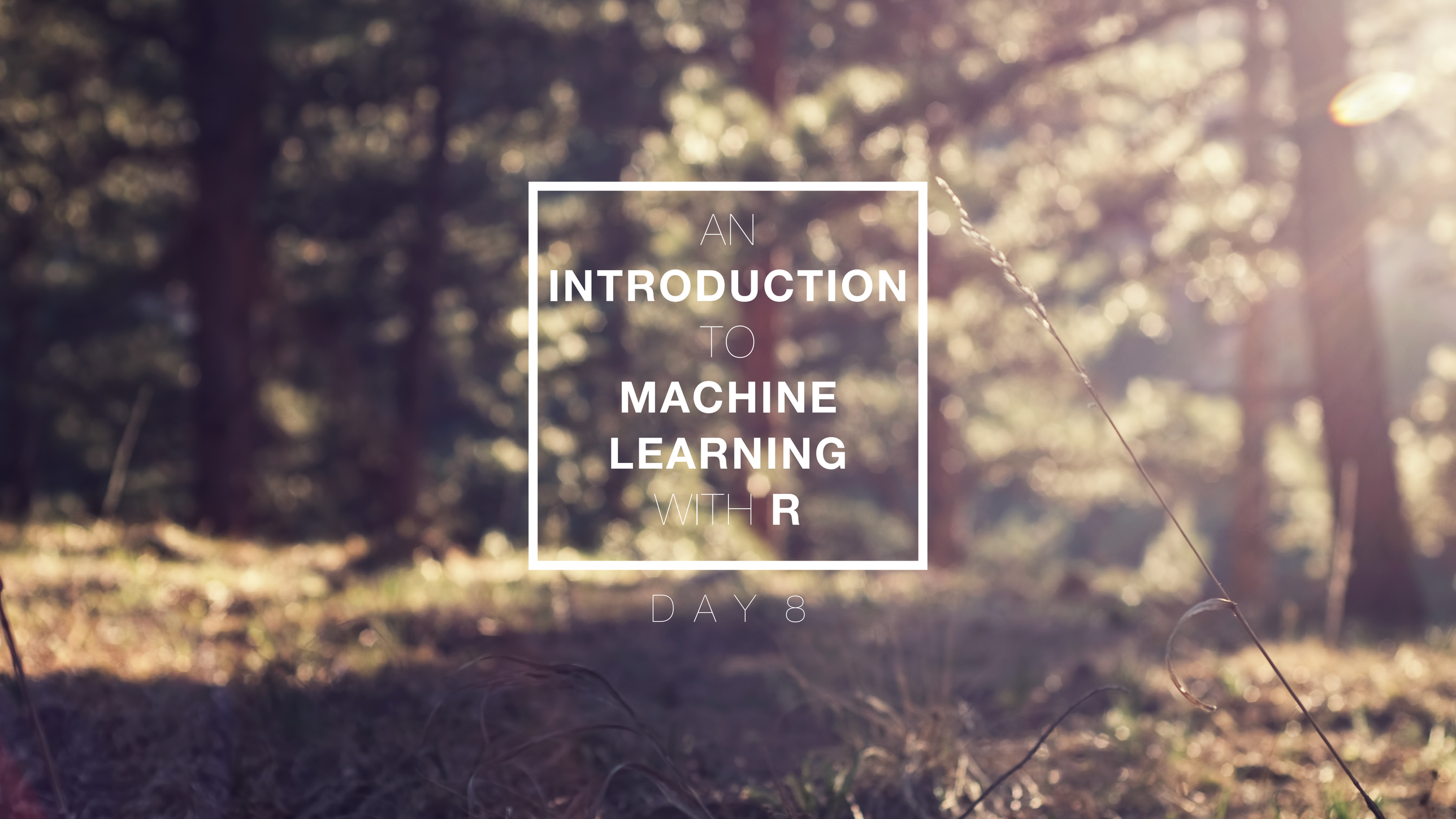
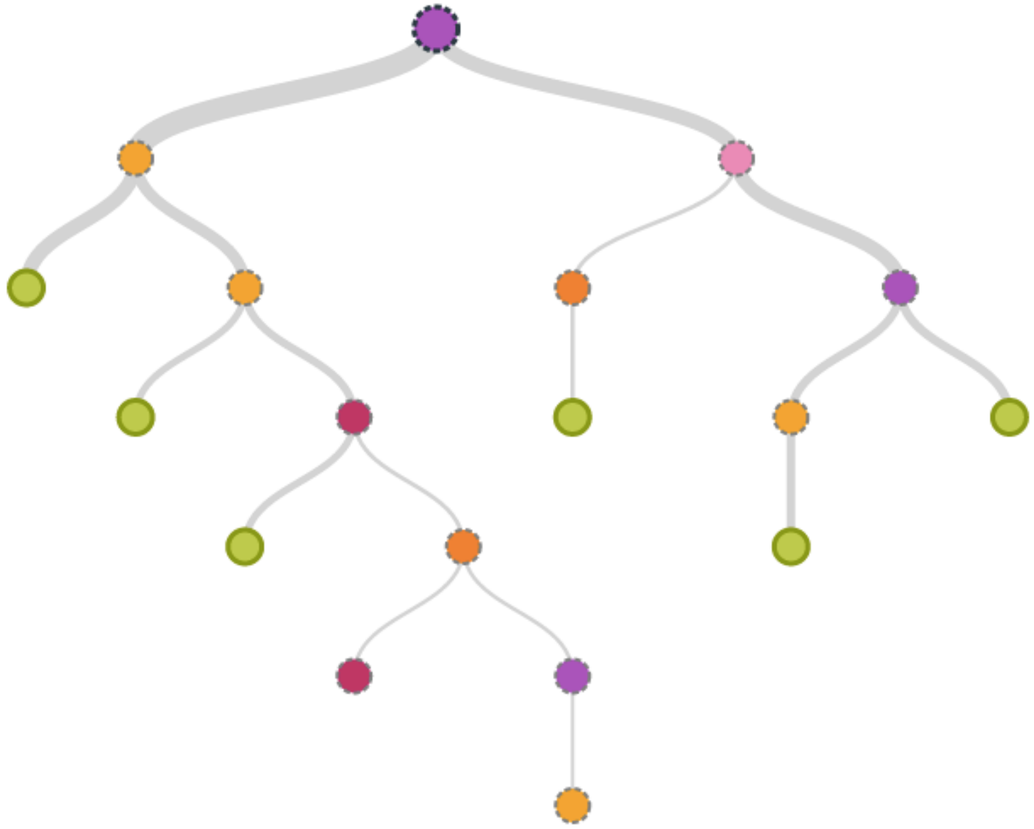

AN
INTRODUCTION
TO
MACHINE
LEARNING
WITH R
DAY 8

DAY 8

Tree Based Model

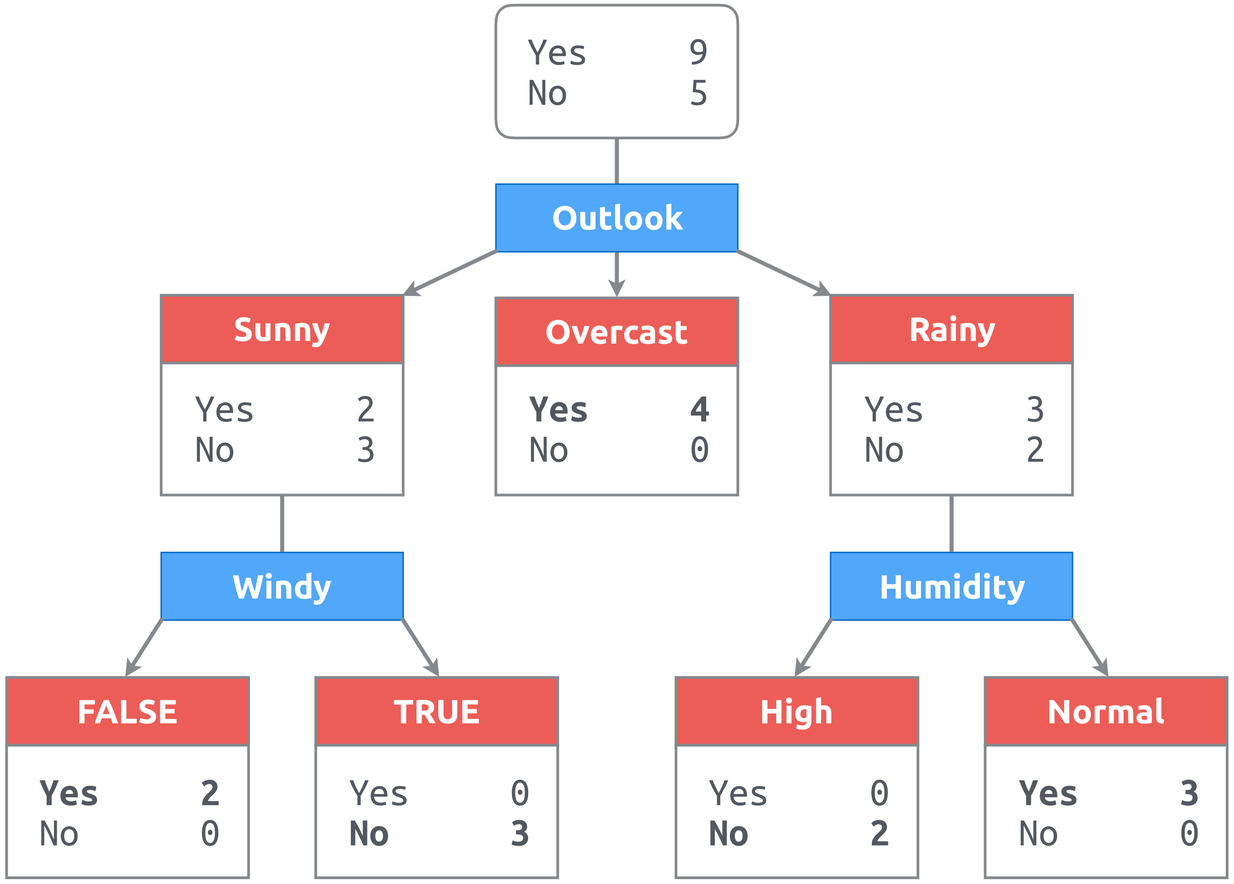
Decision Tree



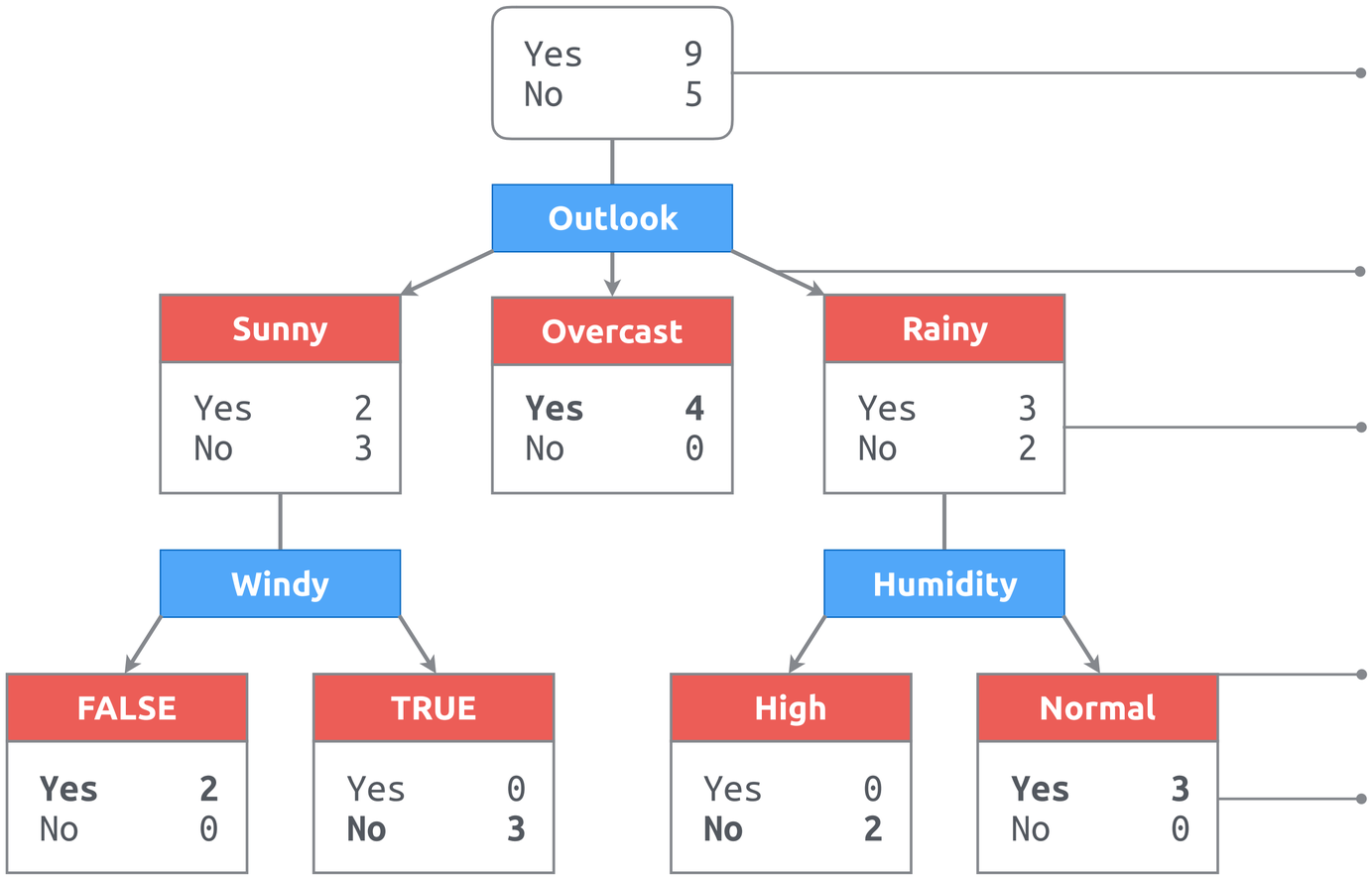
Decision Tree

Predictors — Target

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Overcast	Cool	Normal	TRUE	Yes
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No



Decision Tree



뿌리 노드 (Root Node)
시작하는 마디로 전체 자료를 포함한다.

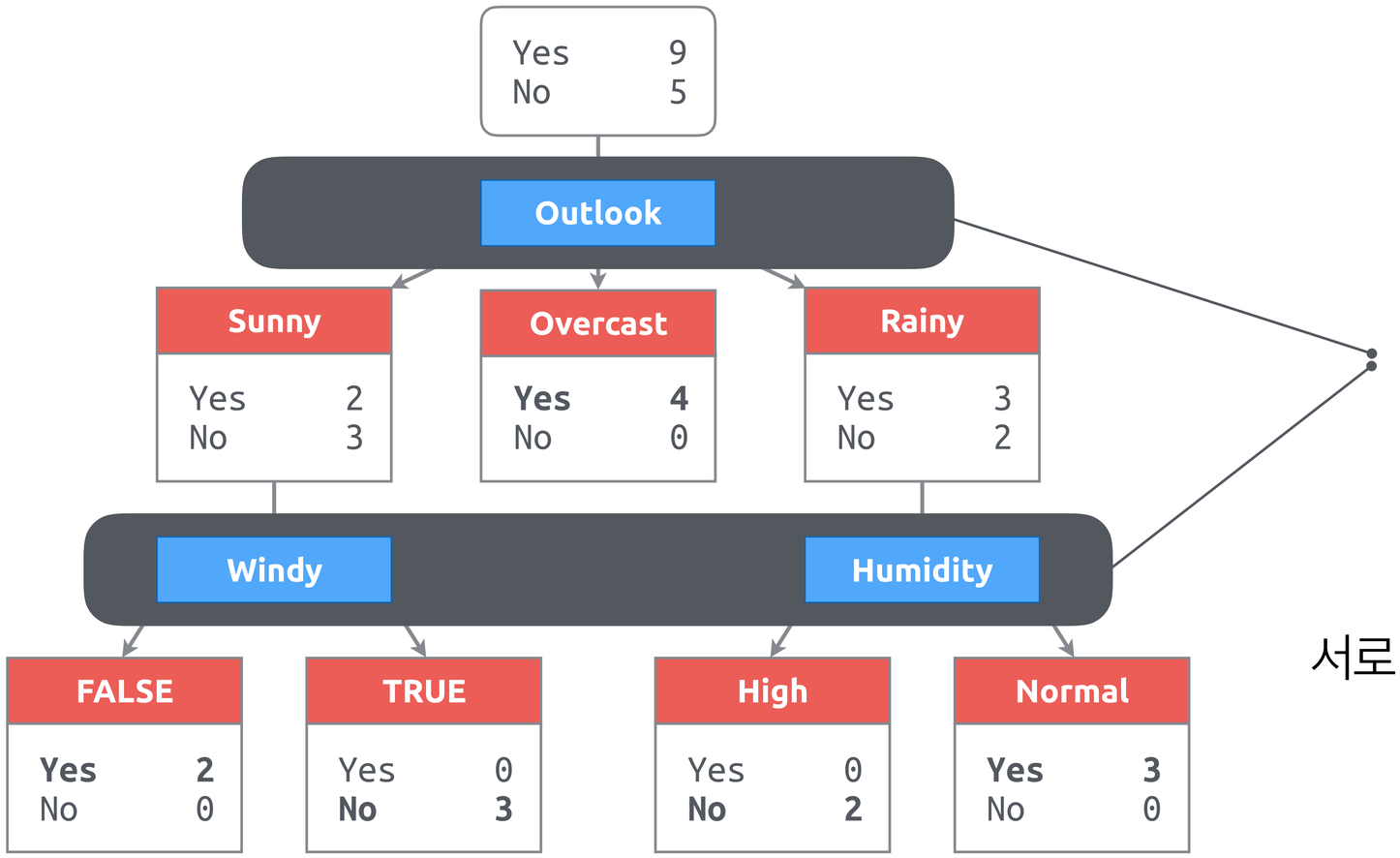
가지 (Branch)

중간 노드 (Internal Node)
뿌리 노드와 잎사귀 노드를 제외한 모든 노드

깊이 (Depth)
뿌리 노드부터 잎사귀 노드까지의 모드 수

잎사귀 노드 (Leaf Node, Terminal Node)
더 이상 밑으로 분리되지 않는 노드

Decision Tree



Split Variable은
무슨 기준으로 뽑힐까

⋮

알고리즘마다
서로 다른 평가지수를 사용한다!

[참고]

Classification **A**nd **R**egression **T**ree

가장 일반적인 사용되는 **"의사결정나무"**

가지를 칠 때, 무조건 두 개씩으로 나눈다.
(이진분류, Binary Classification)

평가지수로 **지니 불순도**를 사용한다.

CART Algorithm

지니 불순도 (Gini Impurity)

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

데이터 파티션

특정 클래스에
포함될 확률

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

CART Algorithm

Age	Income	Student	Credit_rating	buy
Youth	High	No	Fair	No
Youth	High	No	Excellent	No
Middle_aged	High	No	Fair	Yes
Senior	Medium	No	Fair	Yes
Senior	Low	Yes	Fair	Yes
Senior	Low	Yes	Excellent	No
Middle_aged	Low	Yes	Excellent	Yes
Youth	Medium	No	Fair	No
Youth	Low	Yes	Fair	Yes
Senior	Medium	Yes	Fair	Yes
Youth	Medium	Yes	Excellent	Yes
Middle_aged	Medium	No	Excellent	Yes
Middle_aged	High	Yes	Fair	Yes
Senior	Medium	No	Excellent	No

지니 불순도 (Gini Impurity)

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2$$

데이터 파티션

특정 클래스에 포함될 확률

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

$$Gini_{income \in \{low, medium\}}(D) = \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2)$$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right)$$

$$+ \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) = 0.443$$

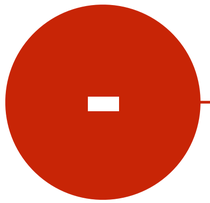
실습

의사결정나무

CREDIT PREDICTION

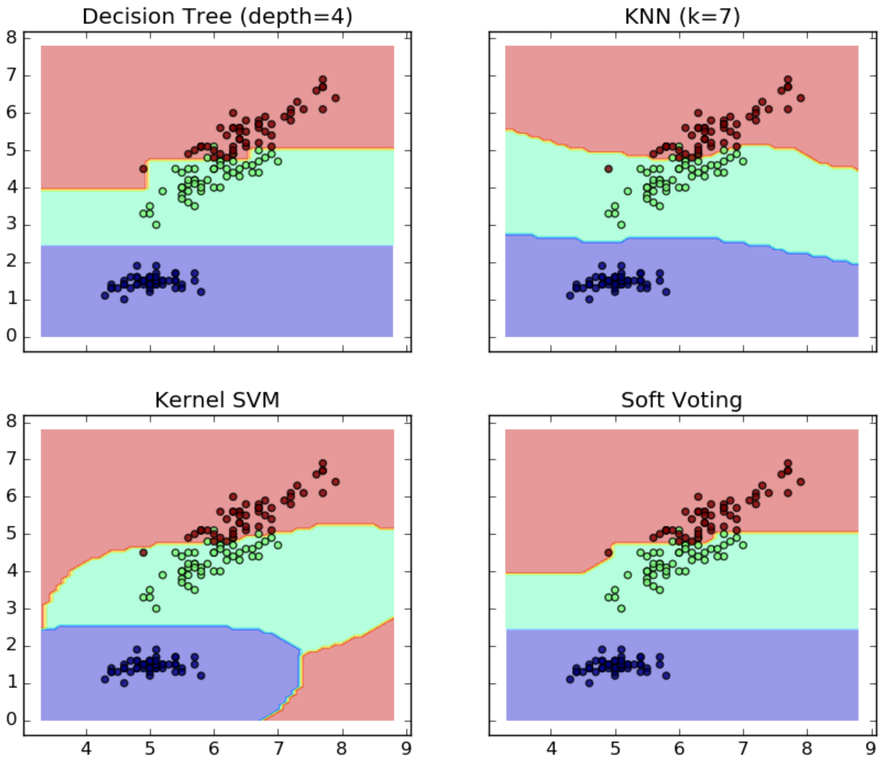
Pros and Cons of Decision Tree

- 노이즈와 이상치(outlier)에 큰 영향을 받지 않음
- 모델의 해석력이 높음
- 연속형 데이터와 이산형 데이터를 모두 다룰 수 있음
- 중요 변수를 채택하기에 용이함



- 과적합 가능성이 높음
- 모델의 유연성이 떨어짐
- 데이터가 적으면 모델을 훈련하기 어려워짐
- 학습 데이터에 따라서 생성되는 의사결정나무에 큰 차이가 생김 (일반성 부족)

Pros and Cons of Decision Tree

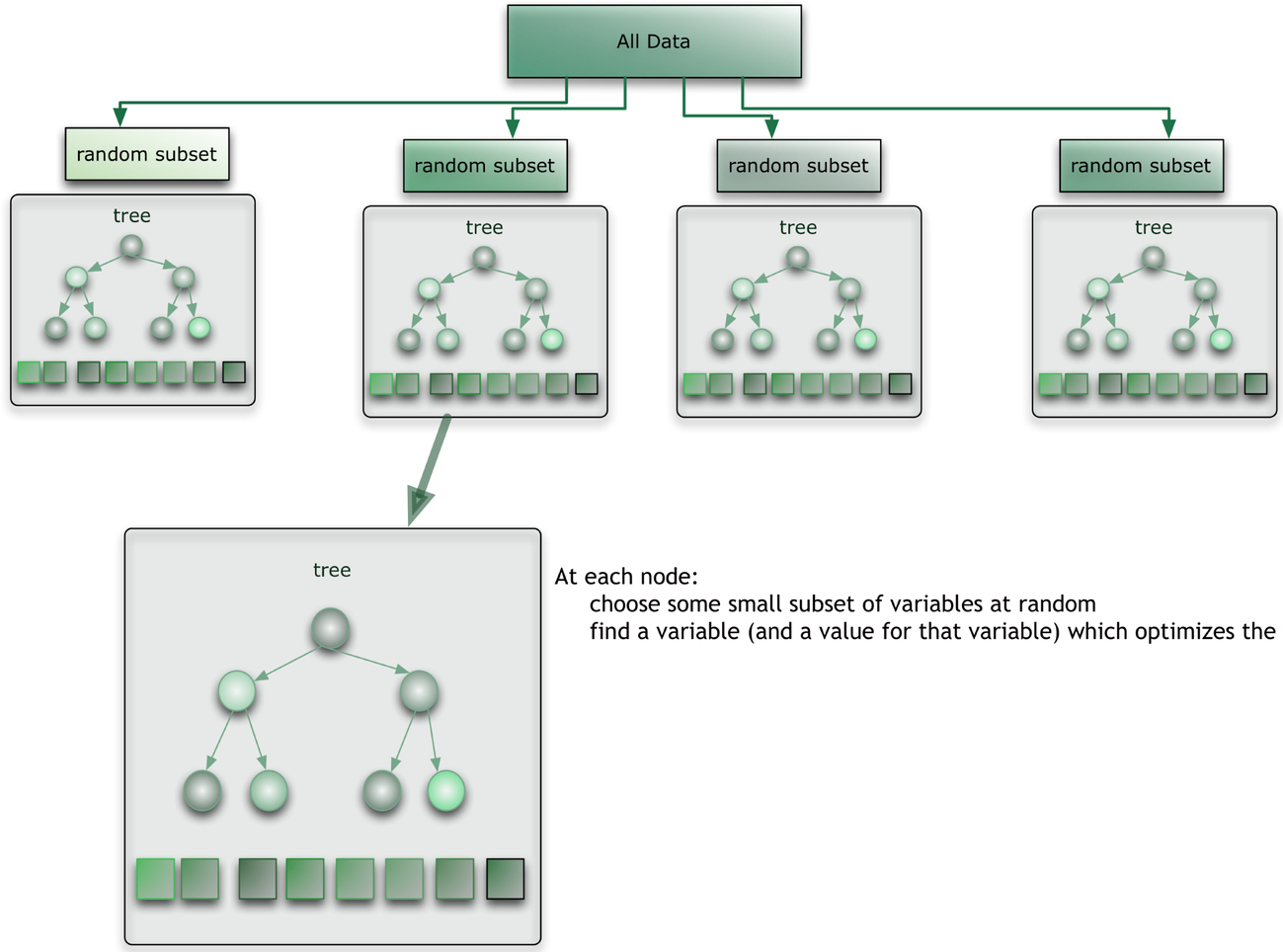


-

- **과적합 가능성이 높음**
- 모델의 유연성이 떨어짐
- 데이터가 적으면
  모델을 훈련하기 어려워짐
- **학습 데이터에 따라서 생성되는**
  의사결정나무에 큰 차이가 생김
  (일반성 부족)

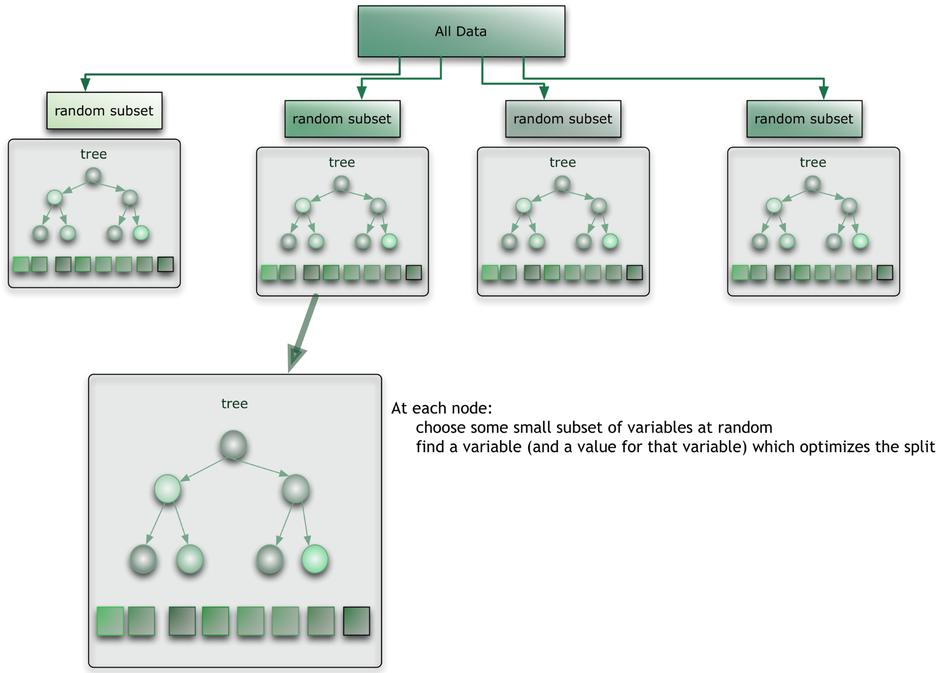
To Overcome This Problems...

Random Forest



At each node:
- choose some small subset of variables at random
- find a variable (and a value for that variable) which optimizes the split

Random Forest

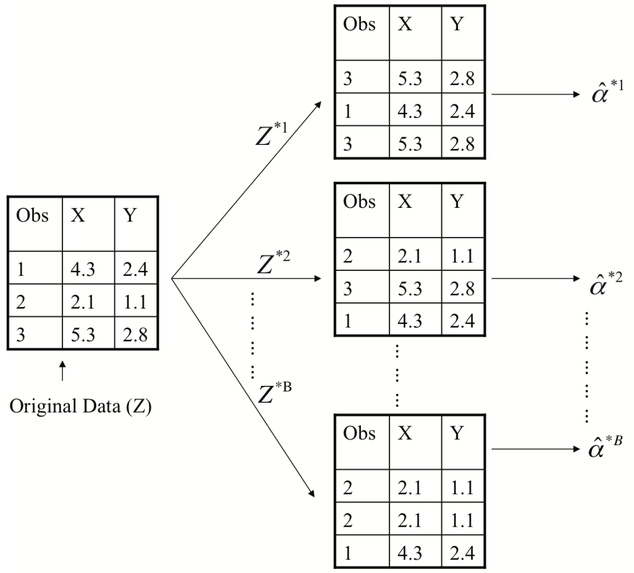


Bootstrapping

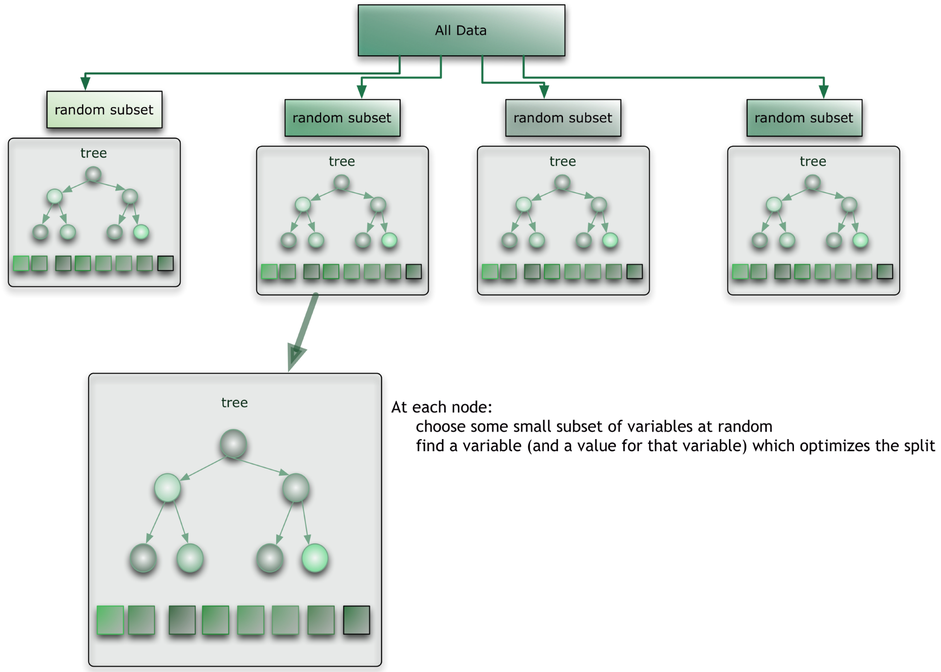
트레이닝 데이터에 대하여

복원표본추출(Sampling with replacement)을 시행하여

새로운 트레이닝 데이터를 만든다.



Random Forest

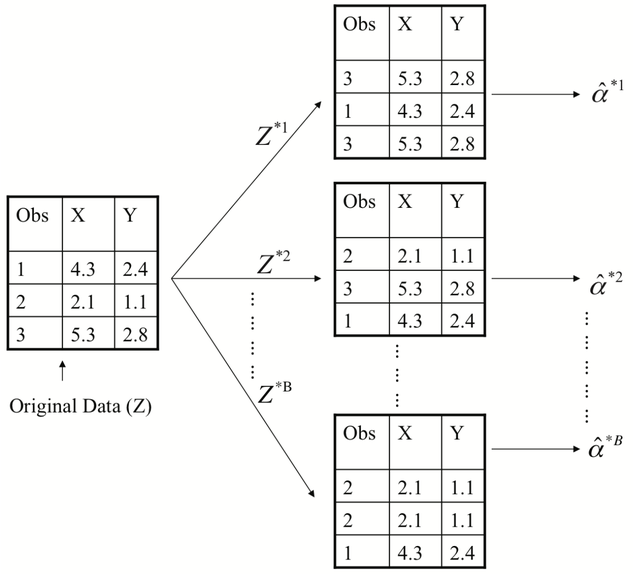


Bootstrapping

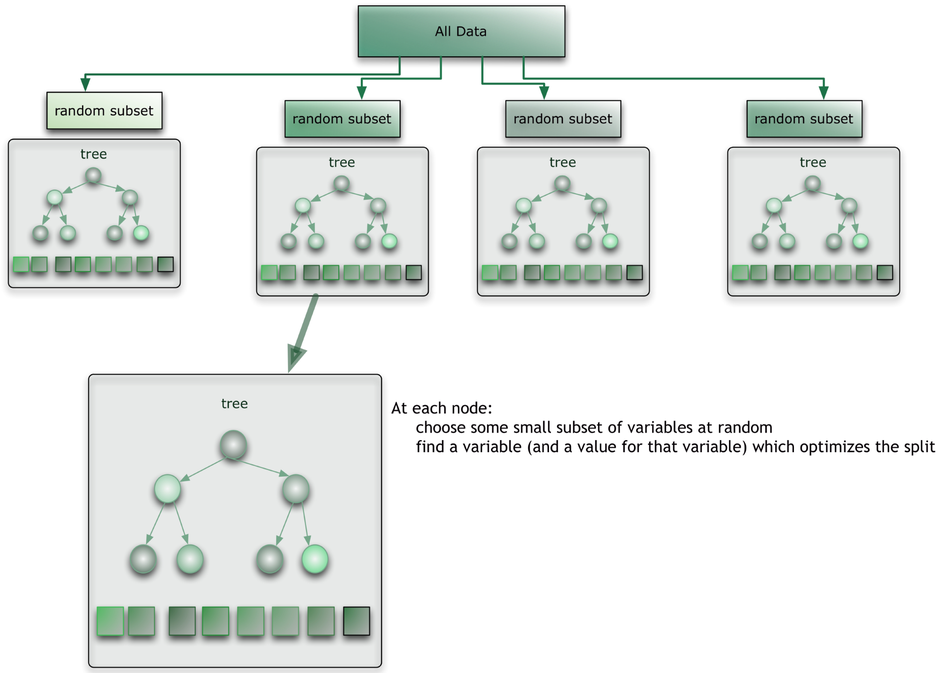
복원표본추출을 하지 않는다면,

기존 트레이닝 데이터 크기의

약 **63.2%**의 샘플 데이터를 얻게 된다.



Random Forest



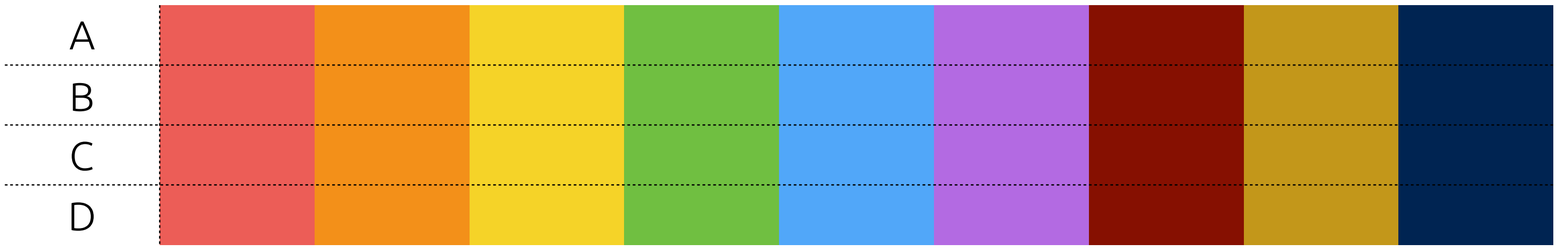
Feature Split

의사결정나무가 가지를 쳐나갈 때마다,

전체 Feature의 개수의 **제곱근** 만큼의

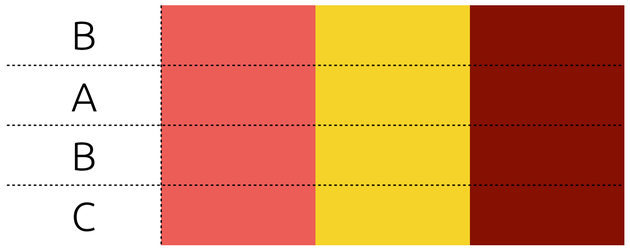
Feature만 가지를 칠 수 있는 변수로 선택된다.

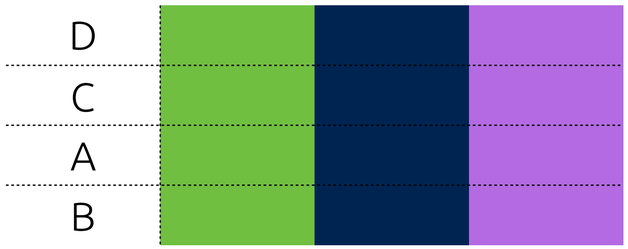
Random Forest

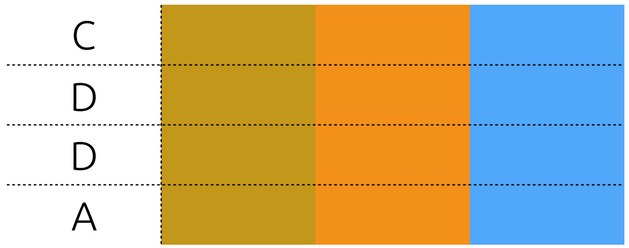


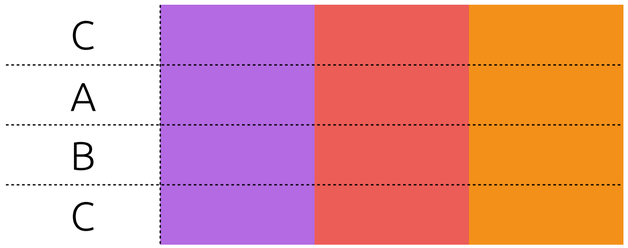
Feature Split

Bootstrapping

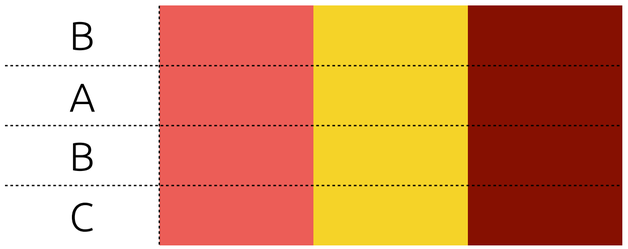


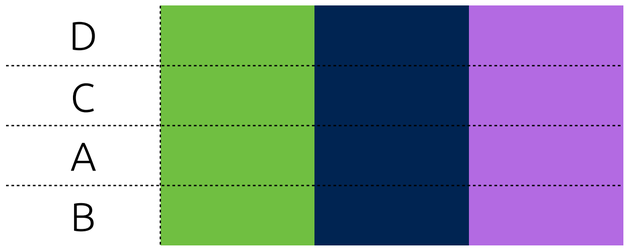


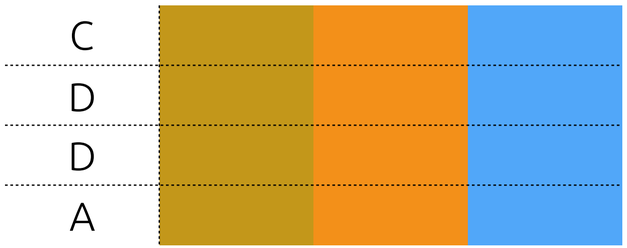


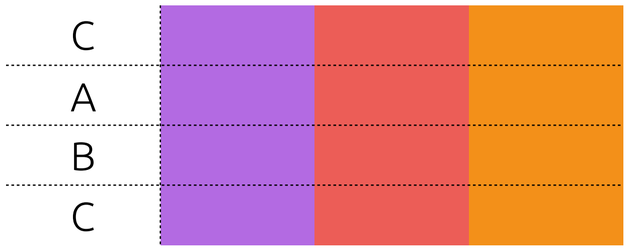


Random Forest



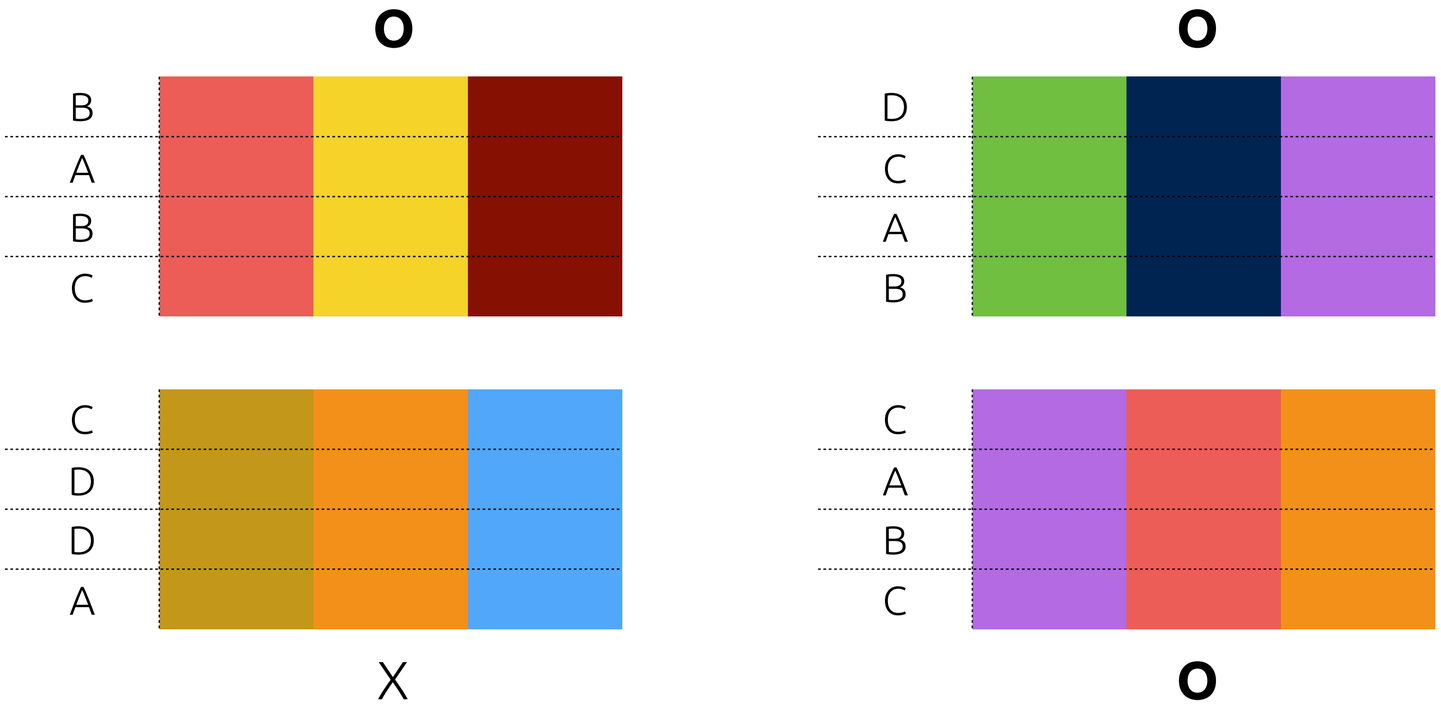






만들어진 데이터 집합들에 대해서
의사결정나무를 생성한다.

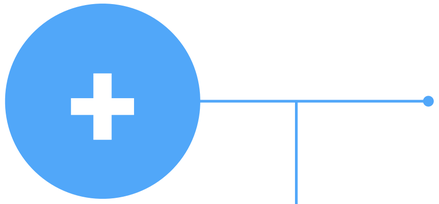
Random Forest



각각의 결과 중 많이 나온 결과값을

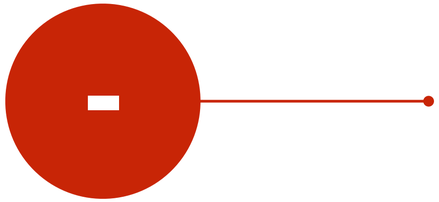
전체 랜덤 포레스트의 결과값으로 반환한다.

Pros and Cons of Random Forest



의사결정나무보다
높은 예측력

일반화가 되어
과적합 가능성이 낮음



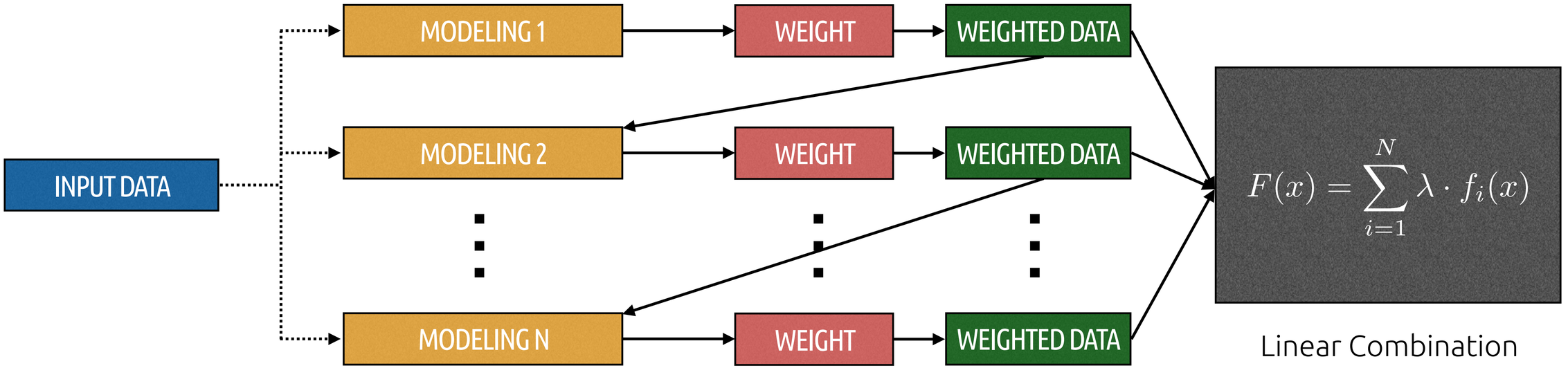
블랙박스 모델
(모델의 설명력이 떨어짐)

실습

랜덤 포레스트

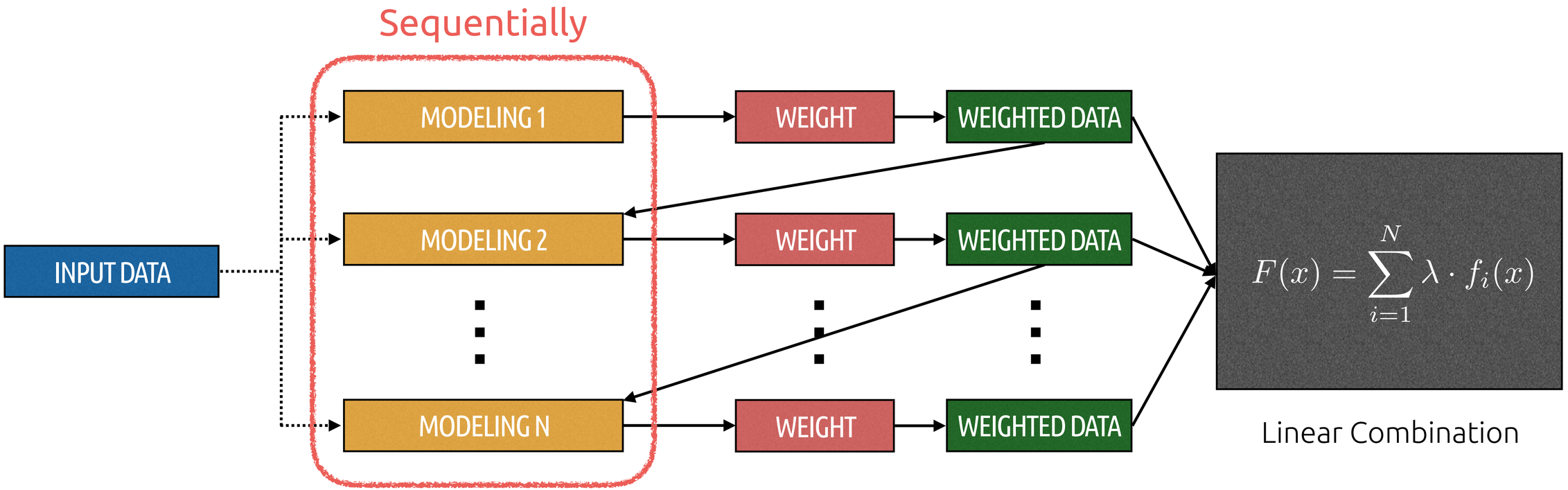
CREDIT PREDICTION

Boosting



트레이닝 데이터에 대하여 모델링을 하여,
잘못 분류된 데이터에 가중치를 주어 새로운 모델을 학습하는 방식

Boosting



트레이닝 데이터에 대하여 모델링을 하여,
잘못 분류된 데이터에 가중치를 주어 새로운 모델을 학습하는 방식

Boosting

Too Many Parameters

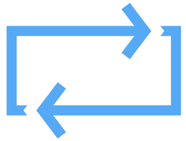
이 값들의 변화는 모델의 성능을 좌우한다.

Learning Rate
λ

학습률은 각각의 모델을
학습하는 비율을 설정한다.

학습률은 기본적으로 낮은 값으로 설정

낮은 학습률 + **많은** 반복횟수
높은 학습률 + **적은** 반복횟수



Iteration
B

반복횟수는 모델을 몇 개를
사용할 지 설정한다.

높은 반복횟수 : **과적합** 가능성 ↑
낮은 반복횟수 : **과소적합** 가능성 ↑

Maximum Depth
D

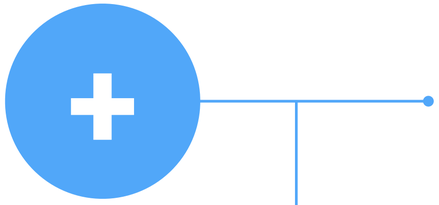
최대 깊이는 각각의 부스팅 트리의
최대 깊이를 설정한다.

각각 부스팅 트리의 복잡도와 관련되어 있다.

의사결정나무에서 **가지치기**하는 이유를 생각하자.

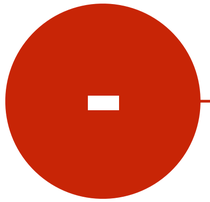
Trade-Off

Pros and Cons of Boosting



의사결정나무보다
높은 예측력

데이터 분석 컴피티션에서
수많은 Winning Solution으로
활용된 바 있음



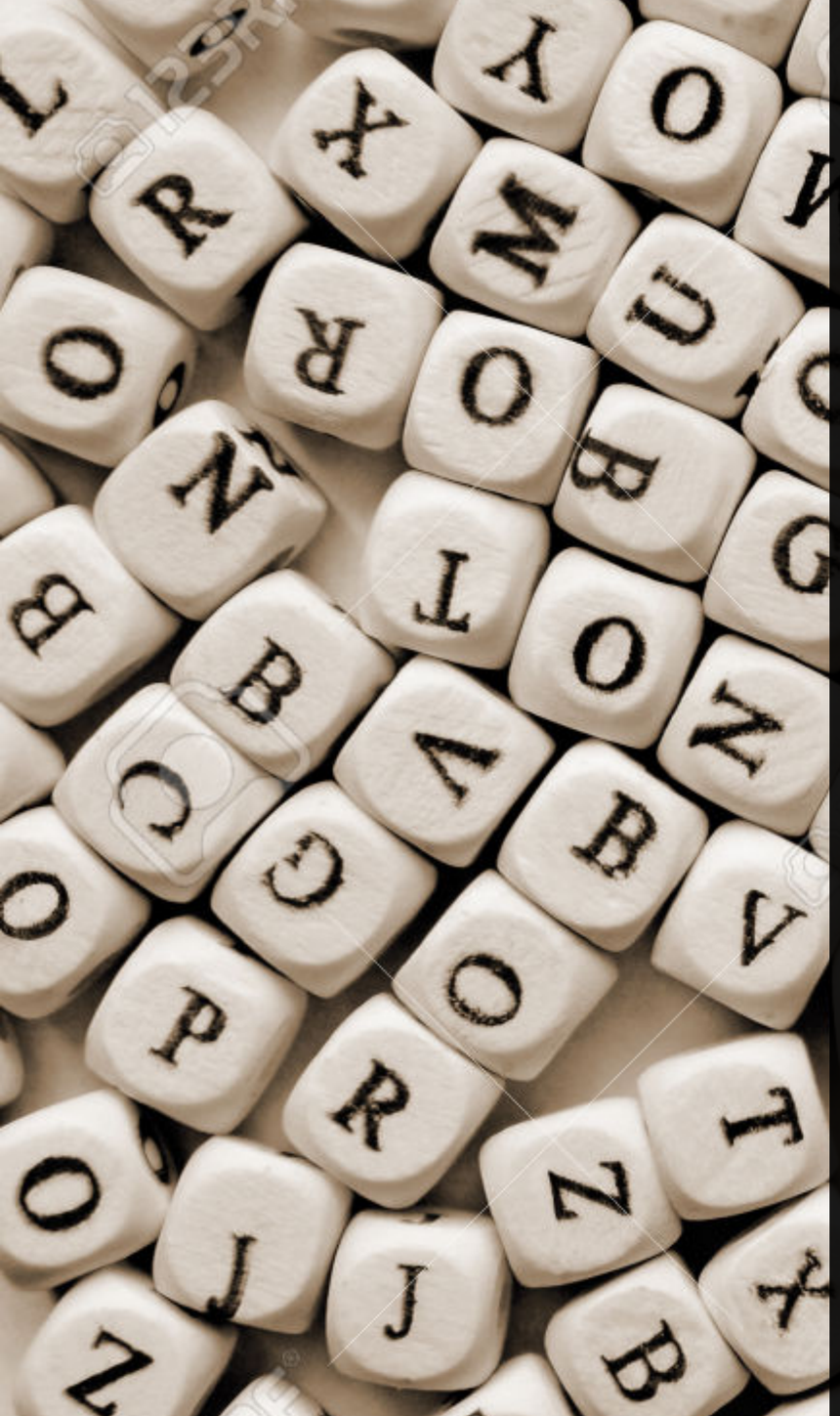
블랙박스 모델
(모델의 설명력이 떨어짐)

파라미터에 따라서
과적합 가능성이 높아짐

실습

부스팅

CREDIT PREDICTION



과제

알파벳 인식

트리 기반 방법을 이용해서
알파벳을 인식해보자.

과제

알파벳 인식

트리 기반 방법을 이용해서
알파벳을 인식해보자.

트레이닝 데이터 : `letter_train`
테스트 데이터 : `letter_test`
데이터 설명 : `letter-recognition.names`

1. 수업 시간에 활용했던 방법론을 사용하여
   주어진 데이터로 알파벳을 인식하는 모델을 만들어보자.
2. 데이터가 크기 때문에
   랜덤 포레스트의 경우는 트리 개수 500개 미만으로 할 것.
   (`ranger` 패키지 참고 가능)
3. 부스팅의 경우, 파라미터 설정이 매우 중요함


THX :)