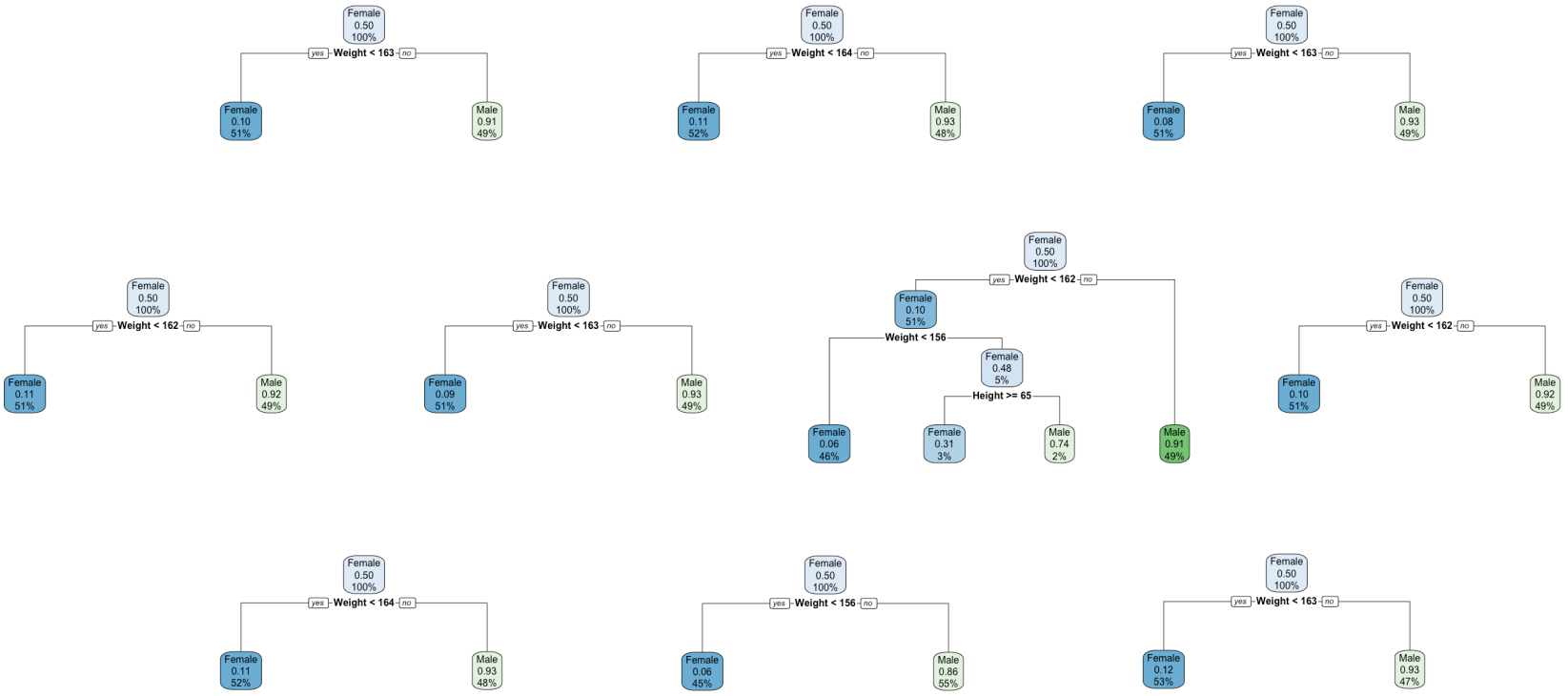
앙상블 메소드

허석진

의사결정나무의 분산

- 훈련 데이터를 다양하게 선택하여 결정나무 생성



표본 평균

- 표본 평균의 분산: σ^2/n
  - σ^2: 모분산, n: 표본의 개수
  - 관측값들의 평균들은 실제 관측값보다 변동이 작아짐

- 중심극한정리
  - "표본 평균들은 정규분포를 가진다"
  - $N(\mu, \sigma^2/n)$

Bootstrapping

- 비복원 추출
  - 훈련 데이터 중 일정 개수의 관측값들을 반복 없이 사용
  - 훈련 데이터가 충분히 많지 않으면
    많은 횟수를 반복하여 비복원 추출할 수 없음
  - 따라서 한 번 사용한 관측값이라도 반복해서 사용할 필요가 있음
- Bootstrapping
  - 훈련 데이터로부터 **복원** 추출하는 방식
  - {0, 1}로부터 5 개를 복원 추출한다면, 예를 들어 {1, 0, 1, 1, 1}

Bagging (Bootstrap Aggregation)

- 연속형 목적 변수 (회귀나무에 적용)
  - 훈련 데이터로부터 표본 복원 추출
  - 각각의 복원 추출한 데이터로 회귀나무 생성
    - 가지치기는 하지 않음 (가지치기를 하면 편향(bias)이 증가)
  - 회귀나무들로부터 각각의 관측값에 대해 예측하고 평균 계산
- 수식
  - Bagging에 의한 예측값 = $\frac{1}{B}\sum_{b=1}^{B}\hat{f}^{b}(x)$
  - B 는 Bootstrap 한 데이터셋 개수
  - $\hat{f}^{b}(x)$는 관측값 x 를 b 번째 데이터셋에 의한 모형으로 예측한 값

Bagging (Bootstrap Aggregation)

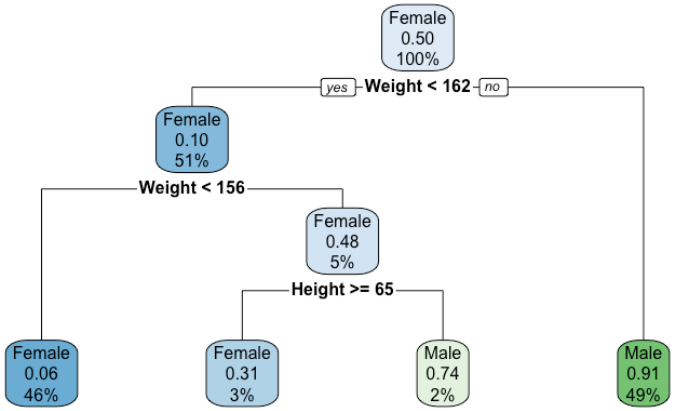
- 범주형 목적 변수 (분류나무에 적용)
  - 훈련 데이터로부터 표본 복원 추출
  - 각각의 복원 추출한 데이터로 분류나무 생성
    - 가지치기는 하지 않음 (가지치기를 하면 편향(bias)이 증가)
  - 분류나무들로부터 각각의 관측값에 대해 예측하고 가장 많이 분류(majority votes)된 범주 선택

Bagging (Bootstrap Aggregation)

- OOB(Out-of-Bag) 오차 측정
  - 표본으로 사용되지 않는 관측값(OOB)들을 테스트에 사용
  - 교차 검증과 유사 (LOOCV 오차와 거의 같음)
  - 교차 검증보다 계산 시간적으로 유리함
- 결정나무와 비교
  - 과적합과 분산 감소 (분산은 1/n로 감소)
  - 조건에 따른 의사 결정 불가능
  - 변수들 사이의 중요도 파악을 위해서는 별도의 계산 필요

Bagging (Bootstrap Aggregation)

- 변수별 중요도
  - 결정나무에서는 분기하는 순서로 변수들 사이의 중요도 판단
  - Bagging에서는 회귀나무와 분류나무 각각에 대해 다르게 계산
    - 회귀나무: 해당 변수에 대한 분기에 의해 오차의 제곱합이 감소하는 정도
    - 분류나무: 해당 변수에 대한 분기에 의해 불순도가 감소하는 정도



Bagging (Bootstrap Aggregation)

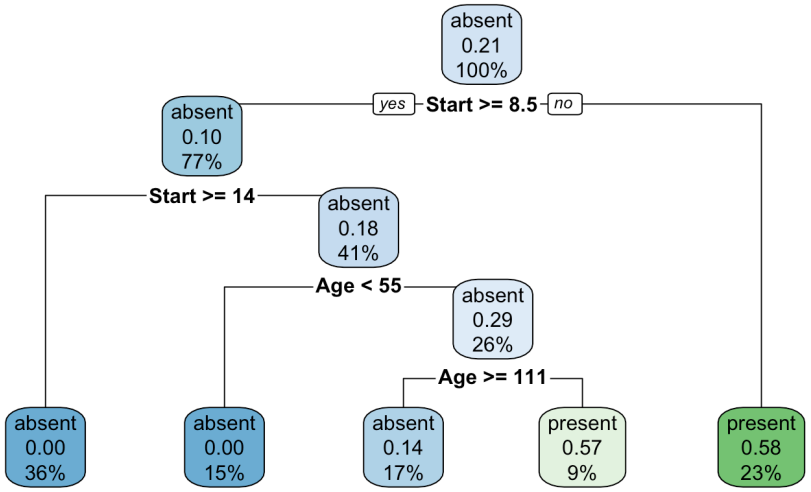
- 모든 변수들이 사용되는 경우의 문제
  - 상대적으로 영향이 큰 변수들이 존재
  - 영향이 적은 나머지 변수들에 대한 훈련 효과가 떨어짐
  - 상관성이 높은 값들을 평균하는 것은 분산을 크게 감소시키지 못함

- 개선
  - 영향이 큰 변수들을 일시적으로 제외하여 다른 변수들의 효과를 정교화
  - **모든 변수가 아니라 일부 변수만 고려하는 것이 나을 수 있음**

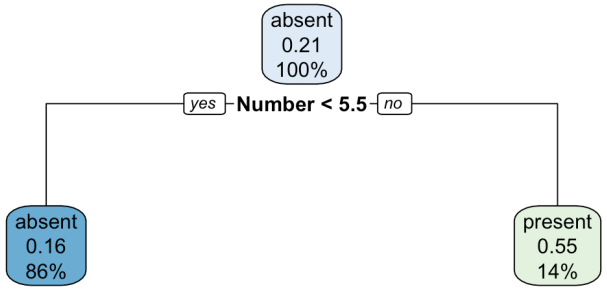
    -> Bagging에서 Random Forest로

Random Forest

- 결정나무에서 가장 영향이 큰 변수를 모형에서 제외



모든 변수 사용



Start 변수 제외

Random Forest 생성

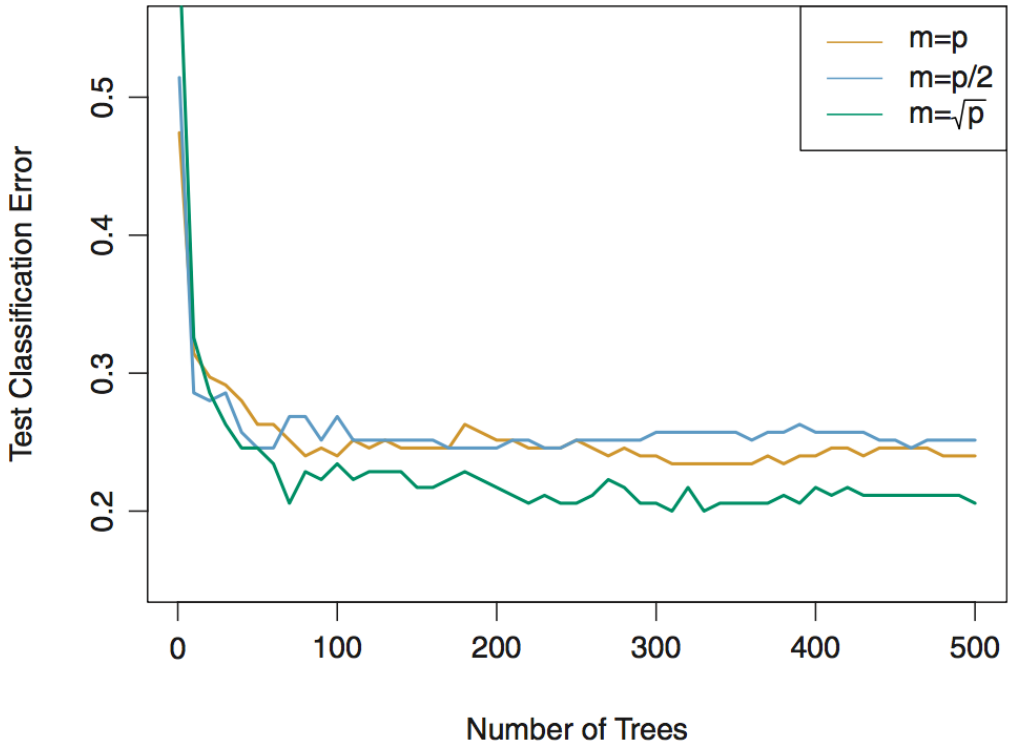
- 훈련 데이터로부터 표본 복원 추출
- **전체 변수보다 적은 수의 변수**를 무작위로 정하여 복원 추출한 데이터로 결정나무 생성
- 예측
  - 연속형: 회귀나무들로부터 각각의 관측값에 대해 예측하여 평균 계산
  - 범주형: 분류나무들로부터 각각의 관측값에 대해 가장 많이 분류된 범주 선택

Random Forest 생성

- 변수의 개수 선택
  - 분류나무의 경우는 $\sqrt{\text{변수의 개수}}$,
  - 회귀나무의 경우 (변수의개수/3) 정도에서 시작해서 조정함
- 장점
  - 과적합과 분산 감소
  - Bagging에 비해 효과적
  - 상대적으로 영향이 적은 변수들에 의한 훈련 효과 증대

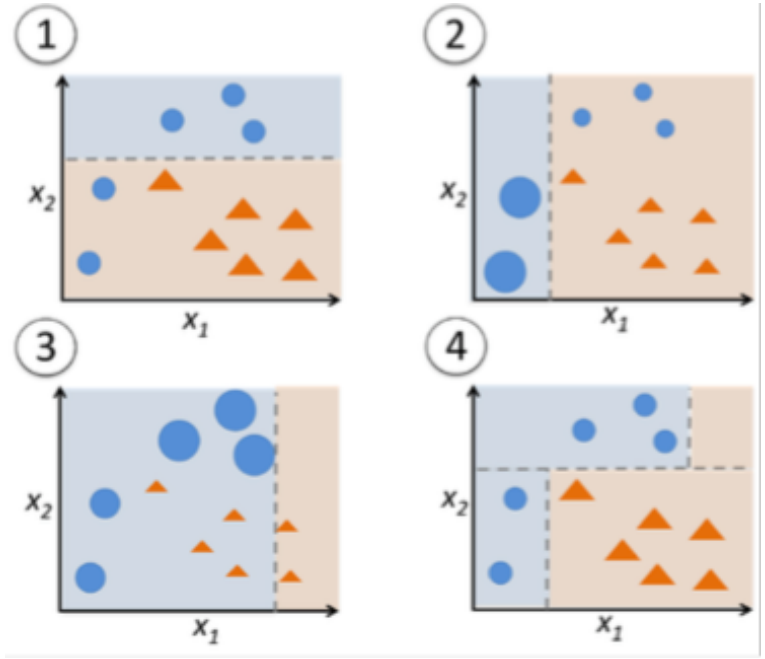
Random Forest 생성

- 나무의 개수에 따른 테스트 오차율



Boosting

- 약한 모형들을 합하여 강한 모형을 만드는 방법



Boosting 알고리듬 (1)

- 모든 관측값들의 가중치가 같은 값에서 시작
- 아래 단계들을 정해진 횟수만큼 반복
  1. 이전 단계에서 잘못 분류된 관측값들에 높은 가중치를 줌
  2. 가중치를 반영하여 새로운 모형 수립
  3. 전체 관측값에 대해 새로 예측

Boosting 알고리듬 (2)

- $\hat{f}(x) = 0$, $r_i = y_i$로 시작
- 아래 단계들을 정해진 횟수만큼 반복 $(\mathrm{b} = 1, 2, \ldots, B)$
  1. 훈련 데이터로부터 모형 $\hat{f}^b$ 적합
  2. $\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$

     $r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$
- $\lambda \hat{f}^b(x)$들을 모두 더하여 boosting된 모형 생성

Boosting 알고리듬 (2)

- 조정 모수
  - 나무들의 개수 B
    - 많을수록 훈련 데이터를 잘 반영
    - 너무 많으면 과적합
  - 수축 파라미터(shrinkage) λ
    - 학습하는 속도
  - 각각의 나무에서 분할의 수