ML Session

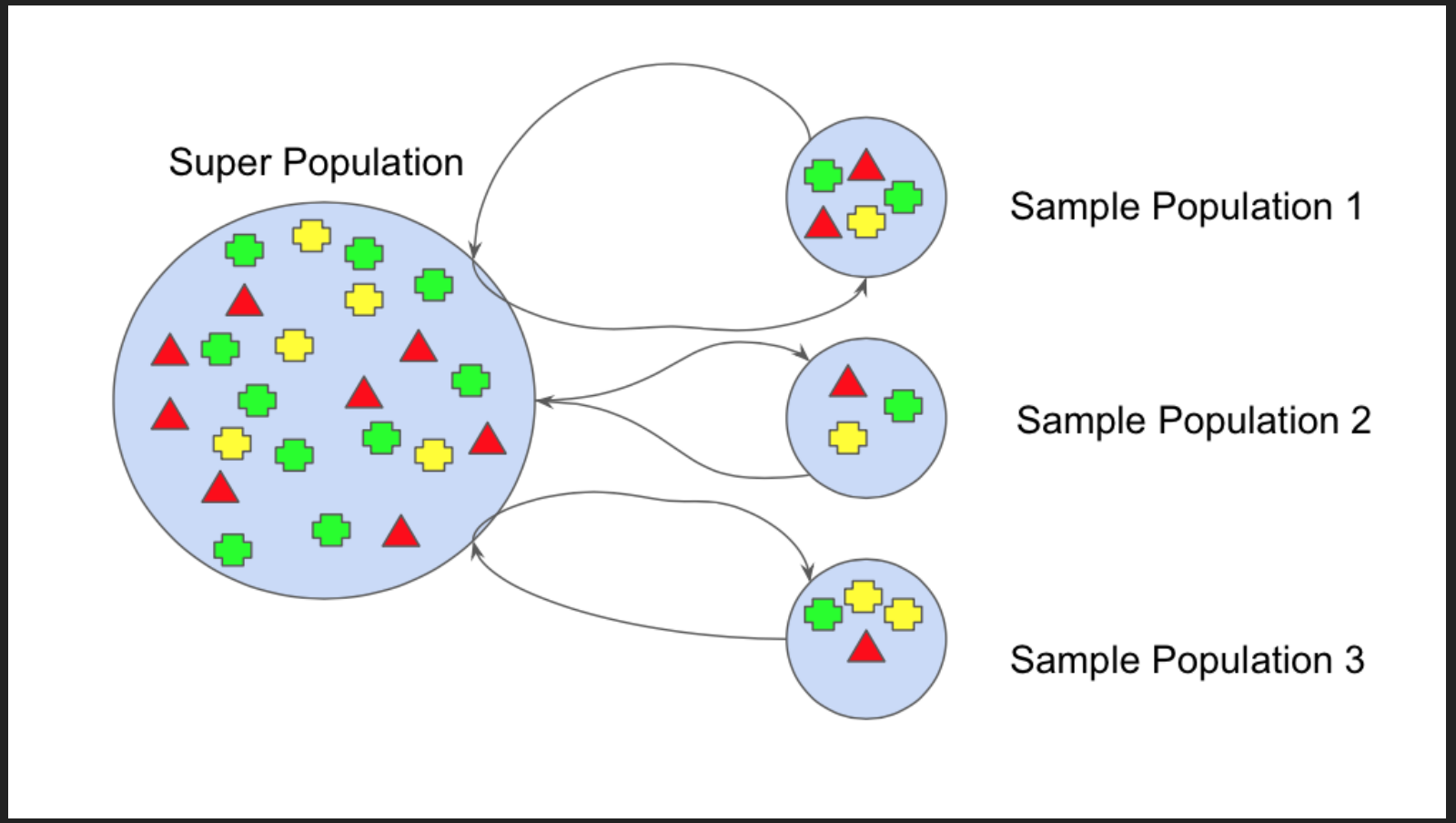
04 Bagging & Randomforest
Confusion Matrix

Bagging??

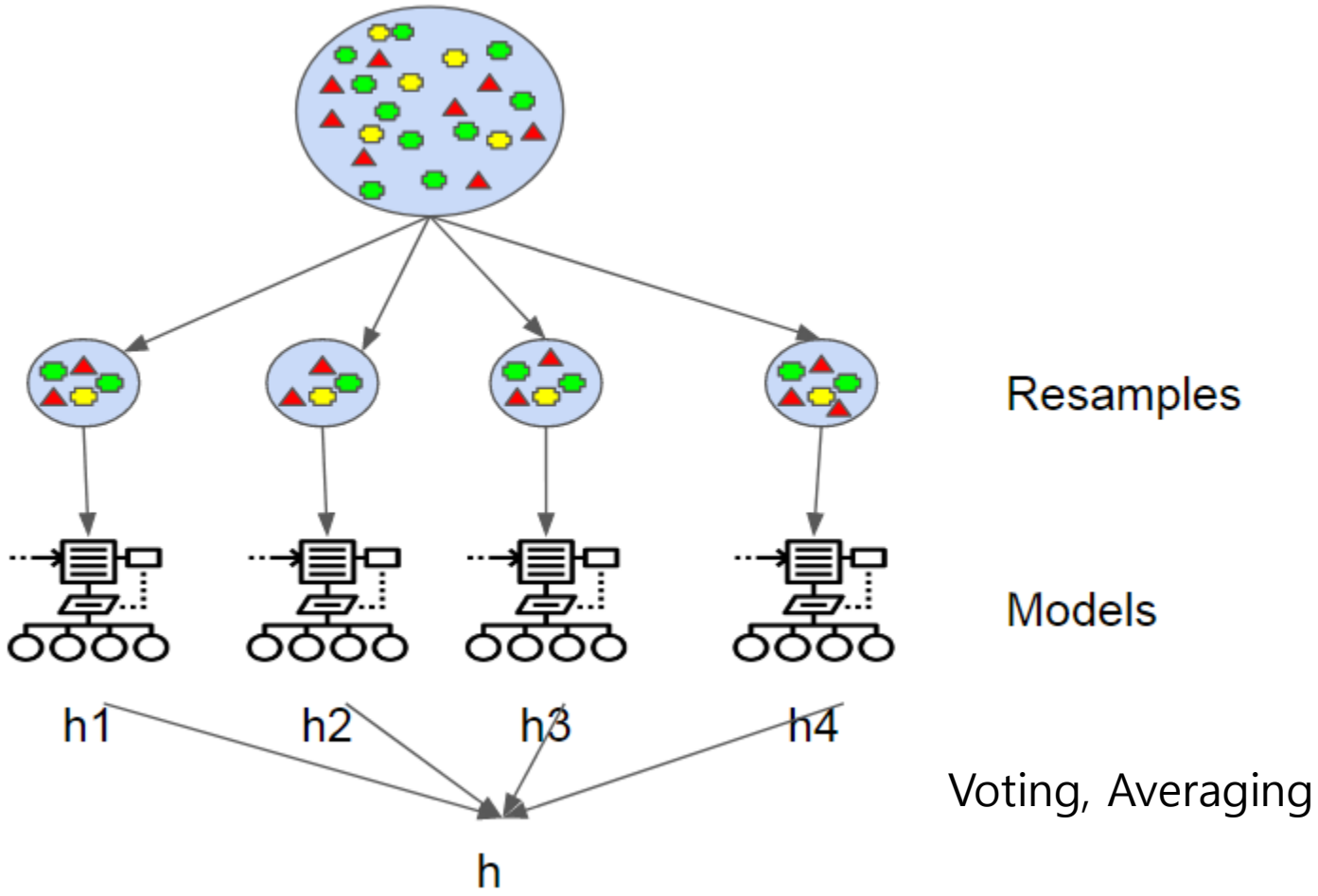
- Bagging : Bootstrap(복원 추출) + Aggregating(집계)
- 같은 모델을 다른 데이터에 적용 한 후 그 결과를 집계하여 결과를 내는 앙상블 기법
- 분산을 줄이기 때문에 과적합 되기 쉬운 모델에 주로 사용

Bagging??

- Bootstraping : 단순 임의 복원 추출



Bagging??



- 결과 집계 과정에서 예측하려는 값에 따라 집계 방식이 달라진다.
- Categorical Data = Voting
- Numeric Data = Averaging

Bagging??

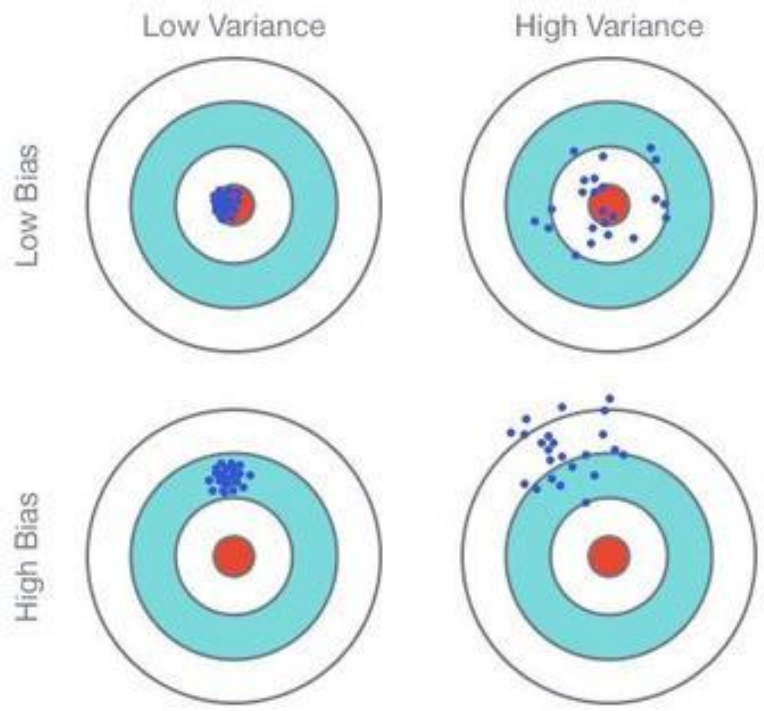


Fig. 1: Graphical Illustration of bias-variance trade-off , Source: Scott Fortmann-Roe., Understanding Bias-Variance Trade-off

- High Variance를 Low Variance로 바꾸는 과정
- 특정 데이터에 과적합된 모델을 일반화 시켜줌

Randomforest

참나무

참나무

참나무

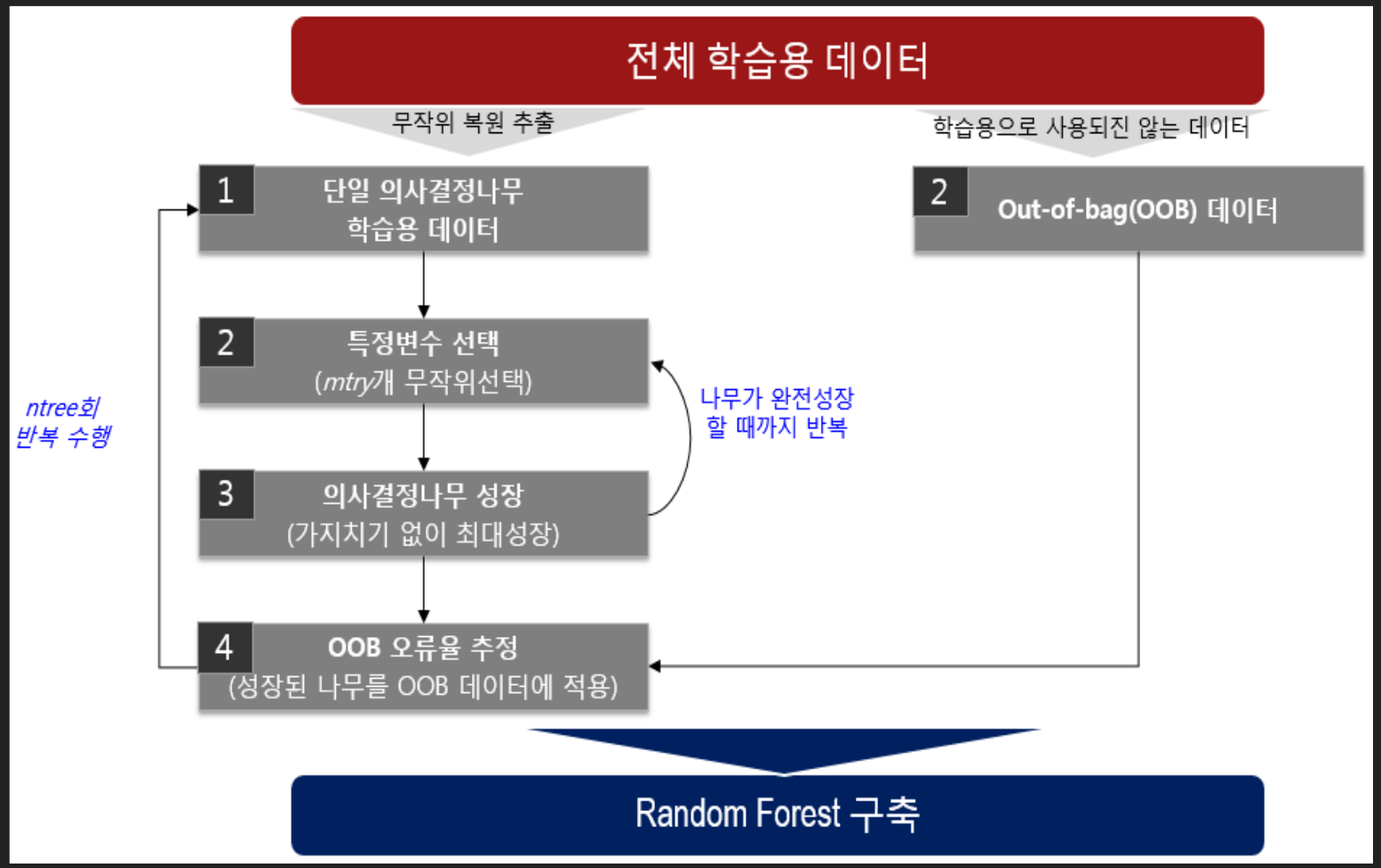
밤나무

참나무

참나무

Randomforest

- DT + bagging + random variable



Randomforest parameters

- ntree = DT의 개수
  caret에서 기본적으로
  제공되지않는 하이퍼 파라미터
- mtry = 무작위 추출할 변수 개수

Randomforest

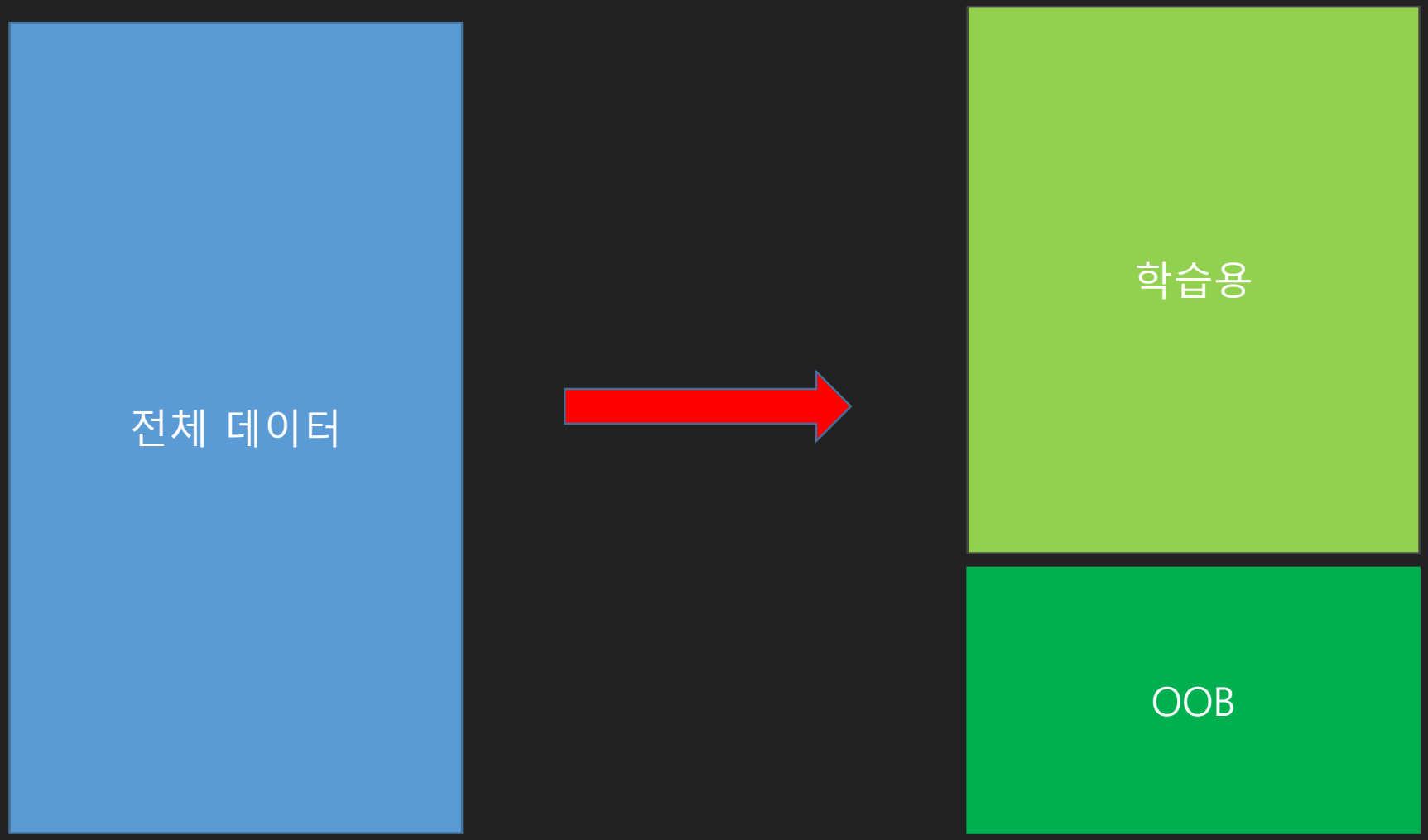
1. 파라미터 선정

Randomforest parameters

- ntree = DT의 개수 (caret에서 기본적으로 제공되지않는 하이퍼 파라미터)
- mtry = 무작위 추출할 변수 개수

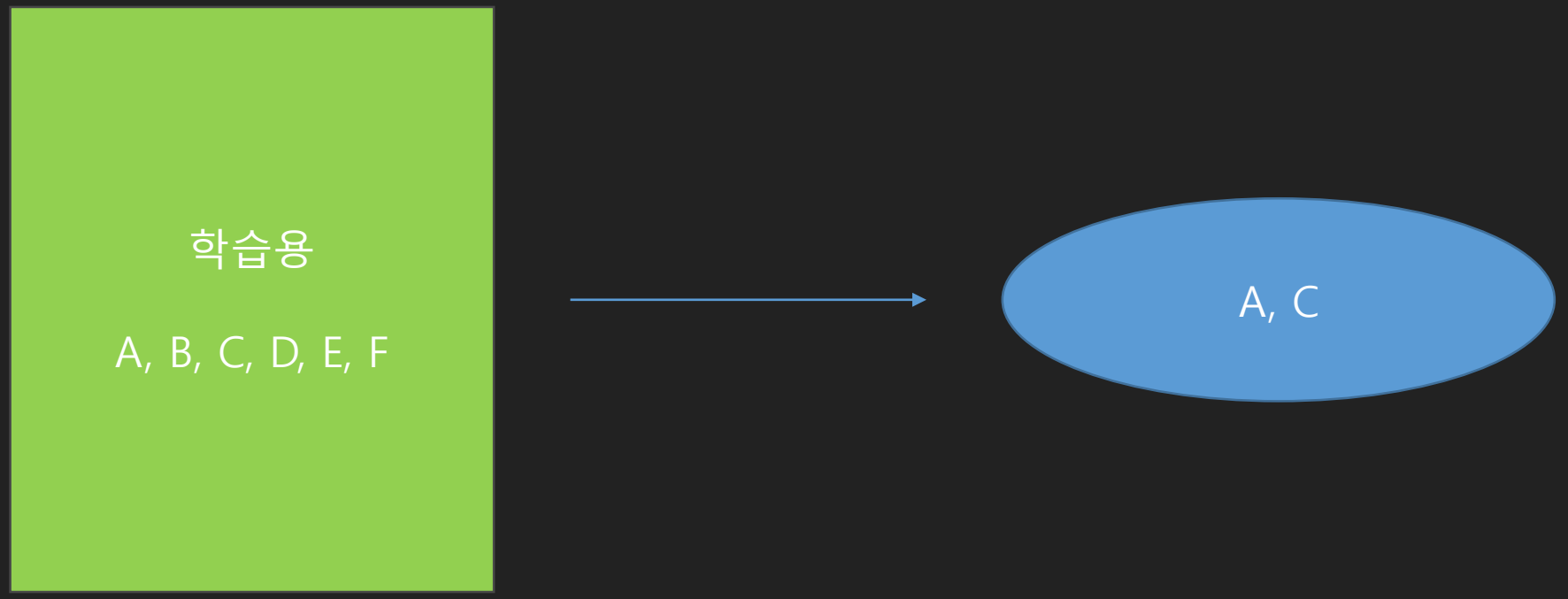
Randomforest

2. 학습용 데이터(2/3)와 OOB 데이터(1/3)로 나눈다.



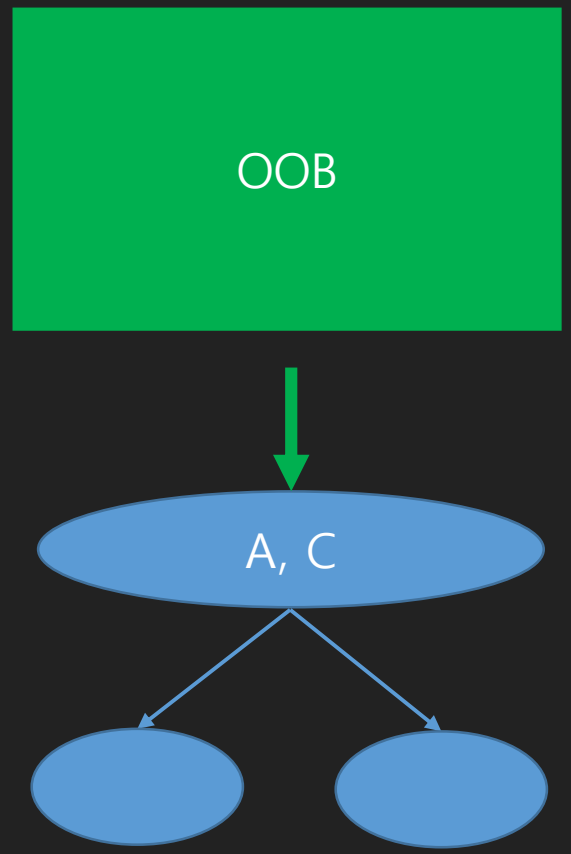
Randomforest

3. 추출한 학습용 데이터에 bagging을 적용하여 데이터를 추출한다.
4. 미리 설정한 mtry 개수만큼 변수를 무작위 비복원 추출한다.



Randomforest

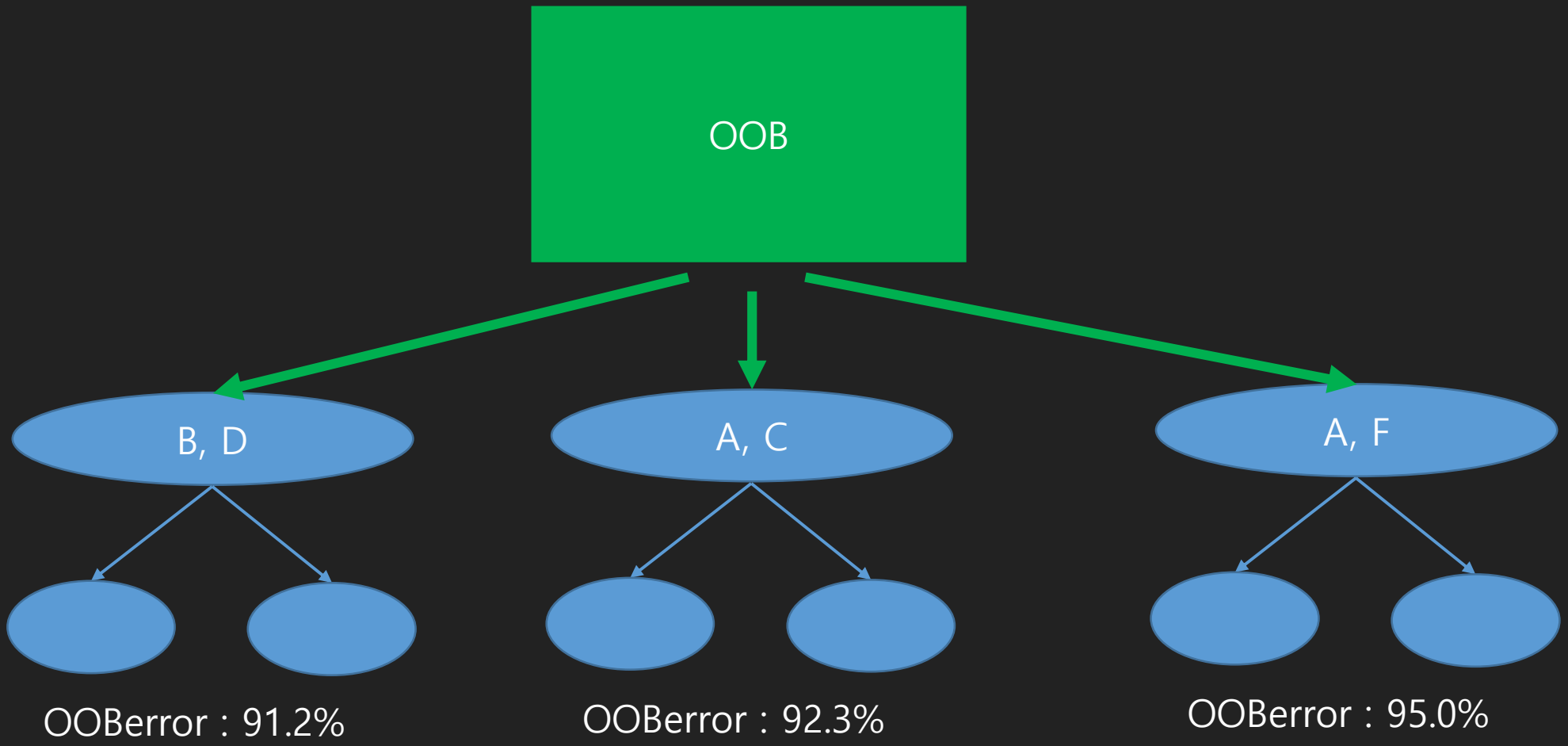
5. DT를 형성하고 OOB데이터에 대한 오류율을 구한다.



OOBerror : 92.3%

Randomforest

6. 1~5번 과정을 ntree번 반복하고 결과를 집계한다.

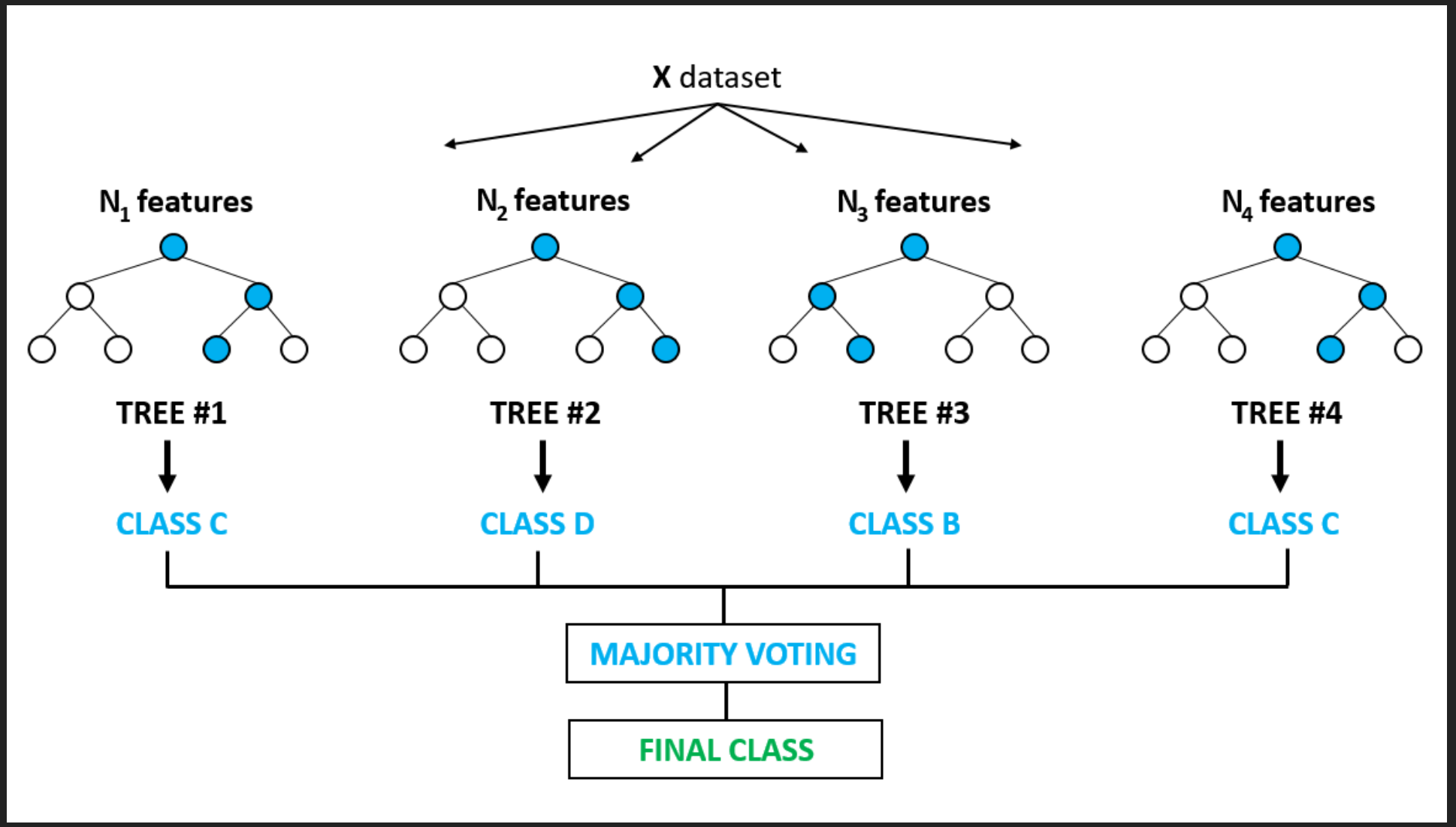


Randomforest

1. 미리 RF에 사용될 파라미터 값을 정한다.(mtry, ntree)
2. 전체 학습 데이터를 2/3은 학습용으로 1/3은 OOB(Out of bag) 데이터로 나눈다.
3. 추출한 학습용 데이터에 bagging을 적용하여 데이터를 추출한다.
4. 미리 설정한 mtry 개수만큼 변수를 무작위 비복원 추출한다.
5. DT를 형성하고 OOB데이터에 대한 오류율을 구한다.
6. 이 과정을 ntree 번 만큼 반복하고 결과를 집계한다.

OOB에러는 모델의 전반적인 성능을 확인할 때도 쓰지만 변수중요도 계산할 때도 쓰인다.

Randomforest



Randomforest

RF의 장점

1. 나무의 수가 많아 질수록 예측 값이 특정 값으로 수렴하기 때문에 과적합을 방지 할 수 있다.
2. 무작위 복원 추출로 인해 Data의 잡음이나 이상치에 영향을 덜 받는다.
3. 변수중요도를 통한 변수 선택이 가능하다.

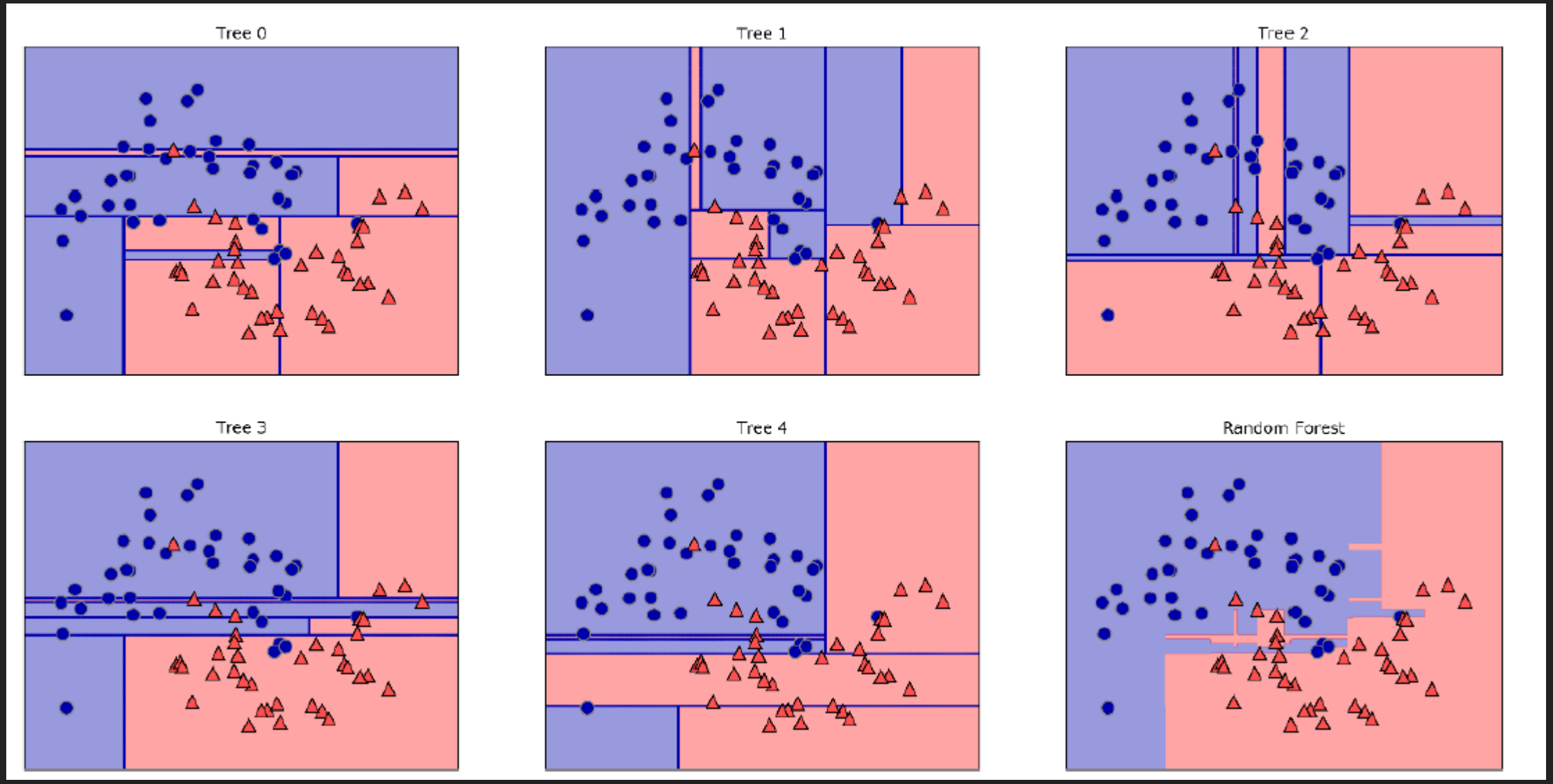
RF의 단점

1. 결과값을 해석하기 어렵다.
2. 변수의 차원이 높고 sparse 한 데이터에는 잘 작동하지 않는다.

Sparse: 데이터가 희박하다는 뜻으로 데이터에 0이나 NA값이 많은 것을 의미함

추후에 PCA기법으로 해결 가능

Randomforest



모형 평가

Accuracy만으로는 모델을 실제 상황에 적용하는 것에 제한이 있다.

건강검진 결과로 암에 양성인지 음성인지 예측하는 모형이 있다면?

양성환자의 비율이 5% 정도라면 그냥 음성으로 찍어도 정확도는 95%가 나온다.

하지만 양성환자를 음성으로 예측하면 음성환자를 양성환자로 예측하는 것과는 비교가 안되는 심각한 문제를 초래할 수 있다.

이런 상황에서 Accuracy는 의미가 없다.

주어진 Task에 맞는 성능지표를 선정해야 한다.

Confusion Matrix

		Predicted		
		Negative (0)	Positive (1)	
Actual	Negative (0)	**True Negative** **TN**	**False Positive** **FP** (Type I error)	**Specificity** $= \frac{TN}{TN + FP}$
	Positive (1)	**False Negative** **FN** (Type II error)	**True Positive** **TP**	**Recall**, **Sensitivity**, True positive rate (**TPR**) $= \frac{TP}{TP + FN}$
		Accuracy $= \frac{TP + TN}{TP + TN + FP + FN}$	**Precision**, Positive predictive value (**PPV**) $= \frac{TP}{TP + FP}$	**F1-score** $= 2 \times \frac{Recall \times Precision}{Recall + Precision}$

일단 이 그림은 외우자

Confusion Matrix

- FP를 줄이는 것이 목표일 때는 precision을 사용한다.
- FN을 줄이는 것이 목표일 때는 recall을 사용한다.
- precision과 recall을 trade-off관계이기 때문에 전반적으로 두 성능지표를 고려하거나 Y의 클래스가 불균형일때는 이 둘을 조화 평균한 f1-score를 사용한다.

Confusion Matrix

우리가 저번주에 사용한 모델의 예측 값은 Y의 클래스였다.
하지만 단순히 예측 값을 클래스로 한다면 정보의 손실이 발생한다. (예측의 불확실성)

예측의 확실성을 얻기 위해서 모델이 예측 값을 반환할 때 확률 값으로 반환 받을 수 있다.

user	Class	P(class = 남)
황태용	남	0.6
남윤주	여	0.1
정재엽	남	0.9
임종언	남	0.75

확률 값을 class로 변환할 때 기준 값을 임계값(thresholds)라고 하는데 위의 문제에서는 0.5이다.
즉 남자일 확률이 0.5이상이면 남자로 분류하고 아니면 여자로 분류하는 것이다.

Confusion Matrix

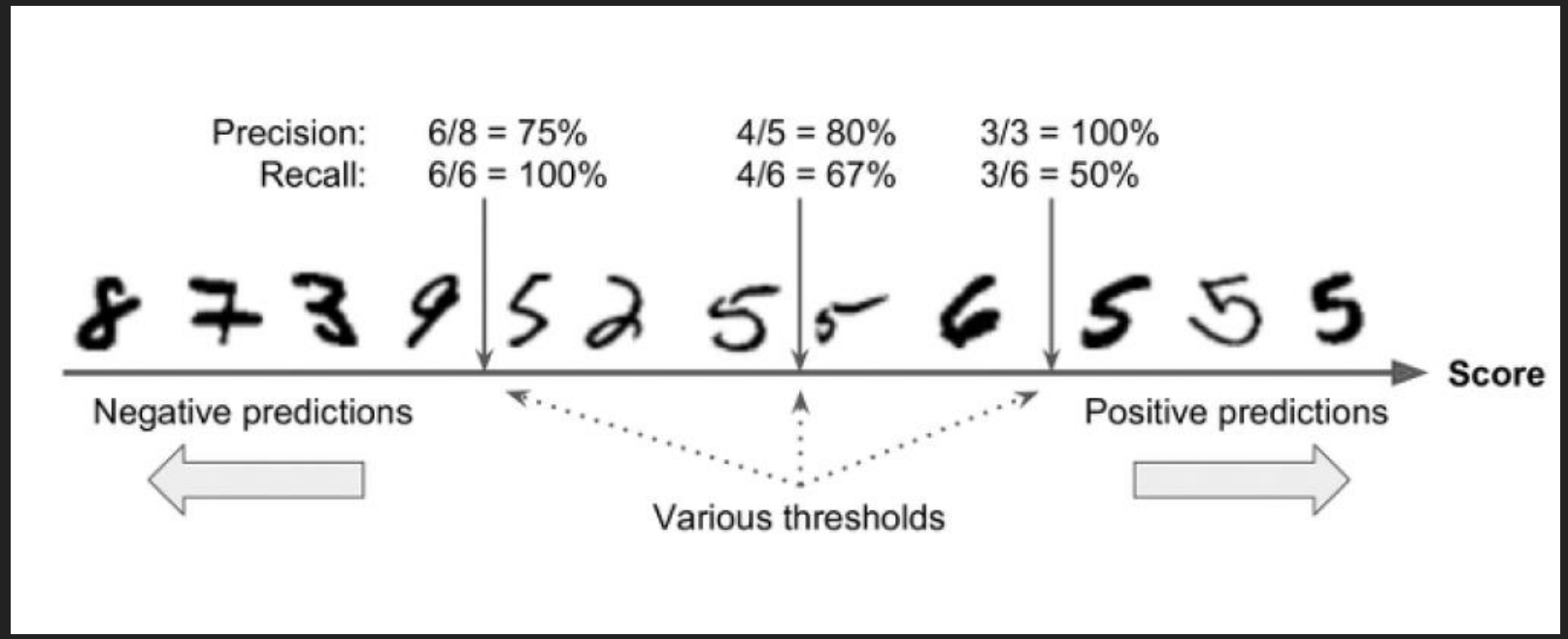
이런 분류 문제에서 내가 만약 더 확실하게 남자인 사람만 뽑기 위해 임계값을 0.7로 올리면 다음 과 같이 바뀐다.

user	Class	P(class = 남)
황태용	여	0.6
남윤주	여	0.1
정재엽	남	0.9
임종언	남	0.75

이런 식으로 주어진 상황에 따라 임계 값을 바꿀 수 있다.

Confusion Matrix

앞서 언급했던 recall과 precision의 trade-off 관계는 임계값 조정으로 인해 발생한다.



손글씨가 5인지 아닌지 분류하는 문제

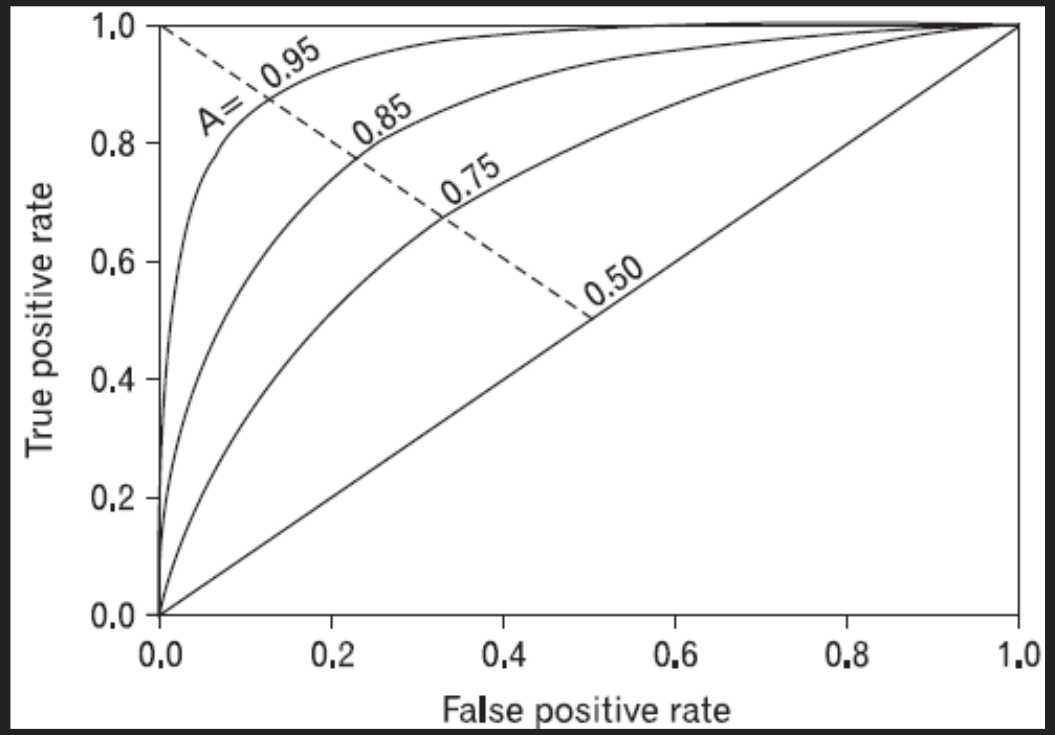
ROC & AUC

- ROC는 sensitivity(recall, TPR)을 y축으로, False positive rate(1-specificity)를 x축으로 놓고 둘 간의 관계를 표현한 그래프

- AUC(area under roc curve)는 말 그대로 ROC curve의 밑의 면적을 계산 한 것이다

  - poor model (0.5~.07)
  - fair model (0.7~0.8)
  - good model (0.8~0.9)
  - excellent model (0.9~1)

- TPR : 1인 케이스에 대해 1이라고 예측한 비율
- FPR : 0인 케이스에 대해 1로 잘못 예측한 비율



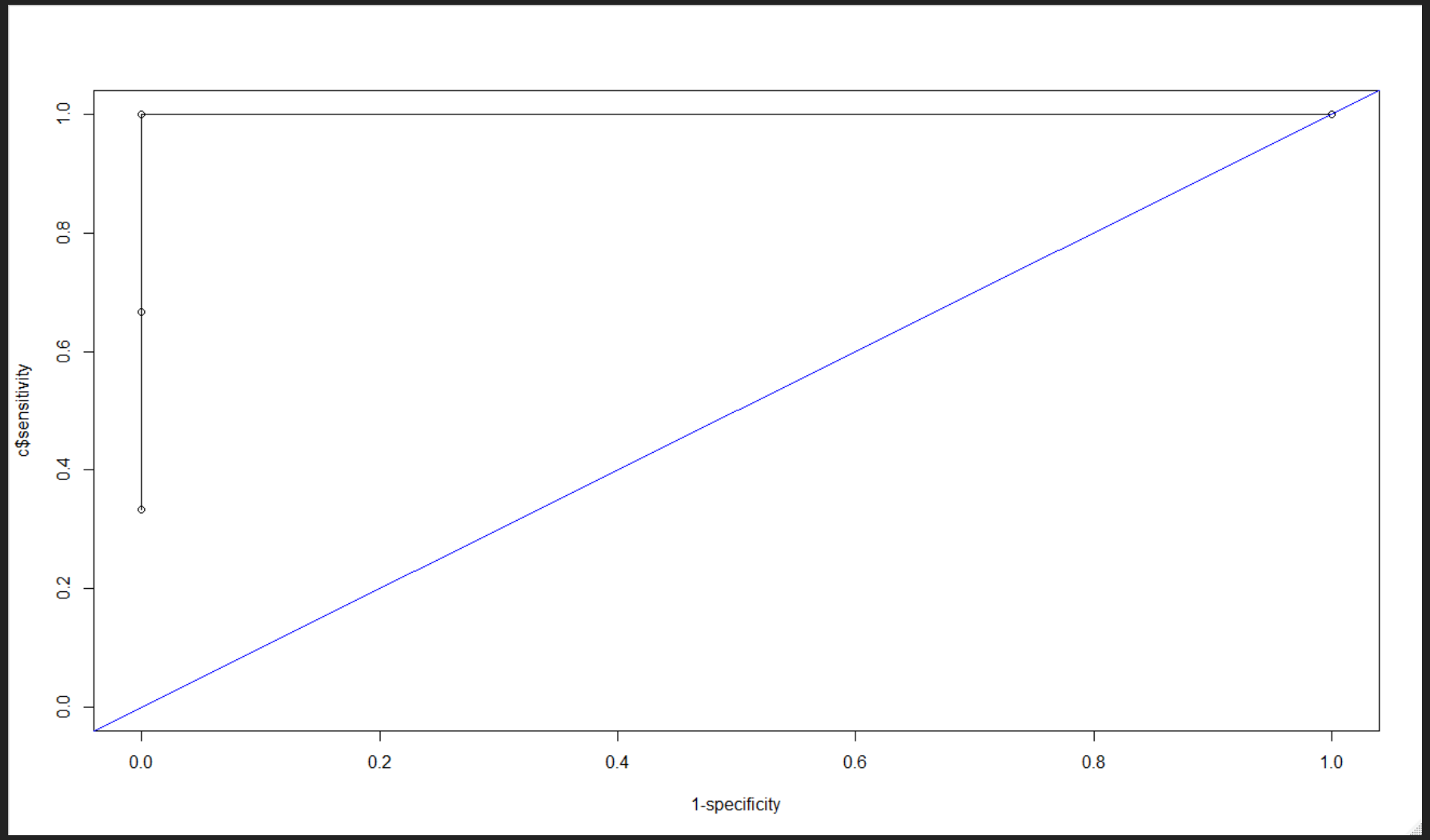
ROC & AUC

- ROC & AUC를 구하기 위해서는 먼저 확률 값이 큰 순서대로 데이터를 정렬한다.

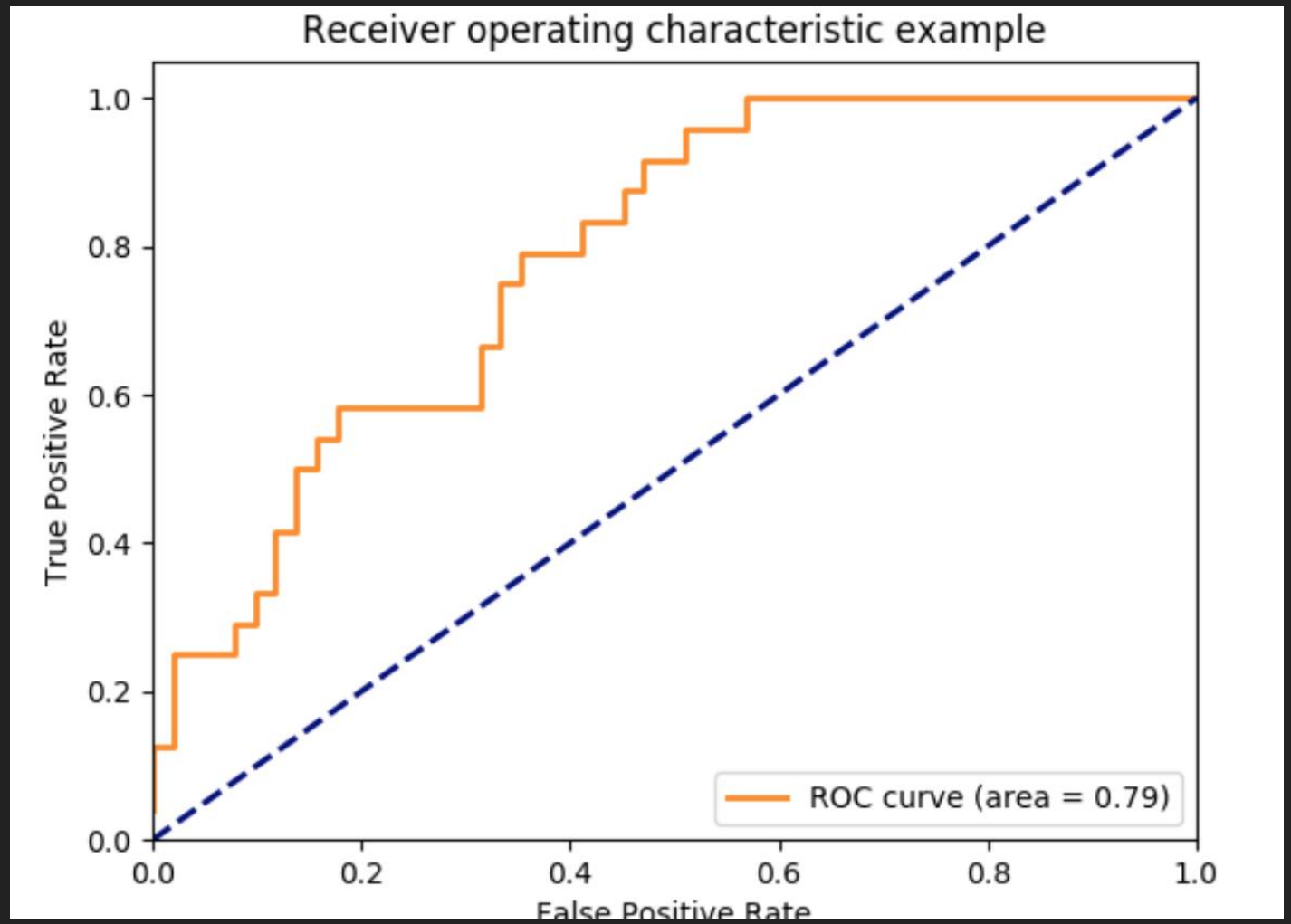
user	P(class = 남)	Class	민감도	1-특이도	특이도
정재엽	0.9	남	1/30	0	1/1
임종언	0.75	남	2/3	0	1/1
황태용	0.6	남	3/3	0	1/1
남윤주	0.1	여	3/3	1	0/1

- 그 후에 임계값을 올려가면서 TPR과 FPR이 만나는 곳에 계속 점을 찍으면 ROC curve가 된다.

ROC & AUC



ROC & AUC



- 임계값이 0이라면?
- 임계값이 1이라면?

과제

1. 랜덤포레스트 구현하기(팀과제, 소스코드 제출(R은 코드를 그대로 제출, 파이썬은 html로 제출))
- 의사결정나무는 패키지를 통해 구현
- bagging 등 앞서 설명한 랜덤포레스트의 의사결정나무와 구분되는 특징들을 구현하는 것이 목적
- 데이터는 실습에 사용했던 암환자 데이터를 사용(결과 값은 암이라고 예측한 확률값을 반환)
- 기존 의사결정나무와의 성능비교(기본 파라미터 사용, 성능이 변화가 없거나 좋지 않아도 상관없음)

2. precision과 recall이 중요한 상황 각각 한 개씩 조사해오기 (개인과제, pdf제출)

QUESTION