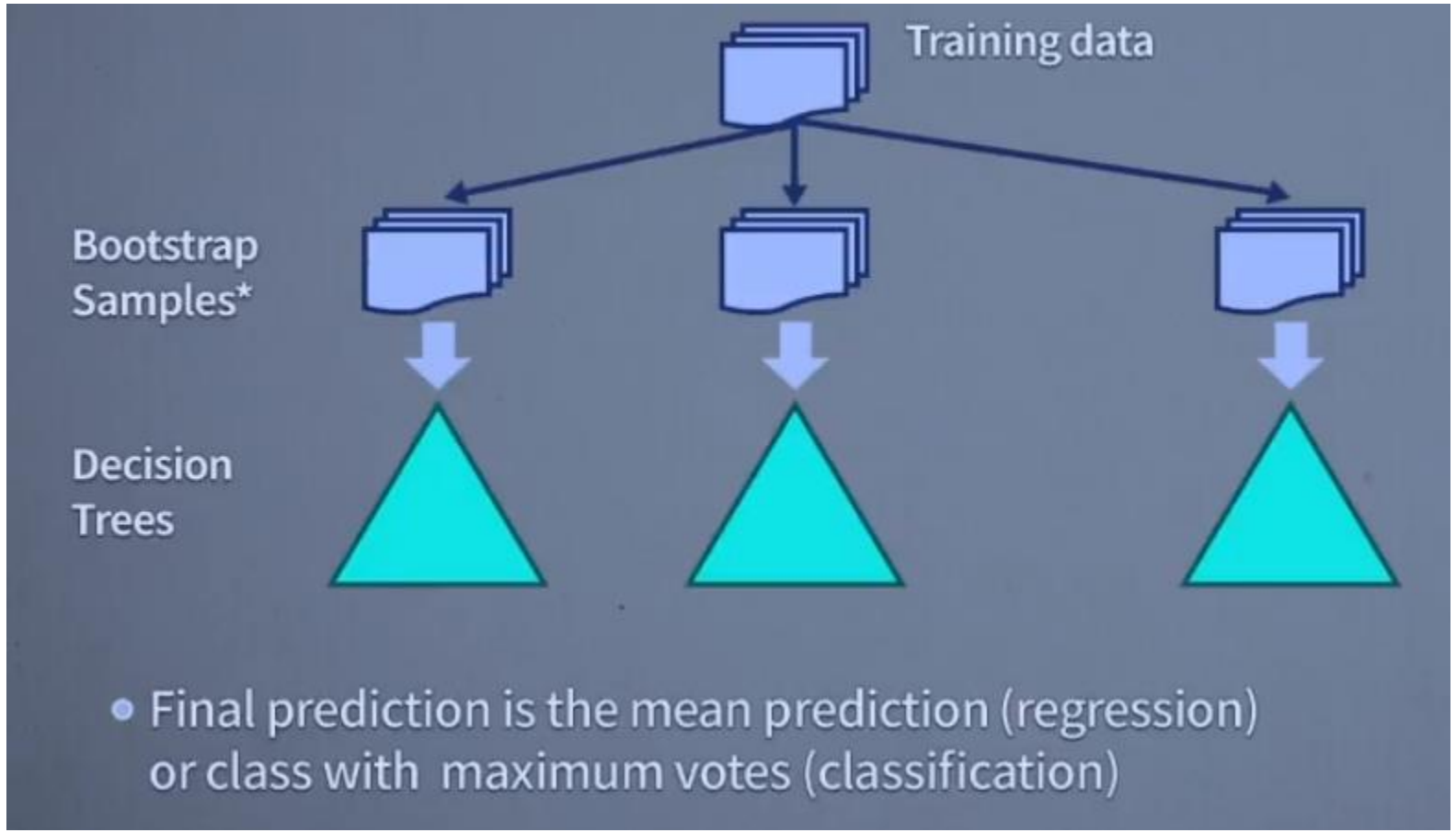
랜덤포레스트

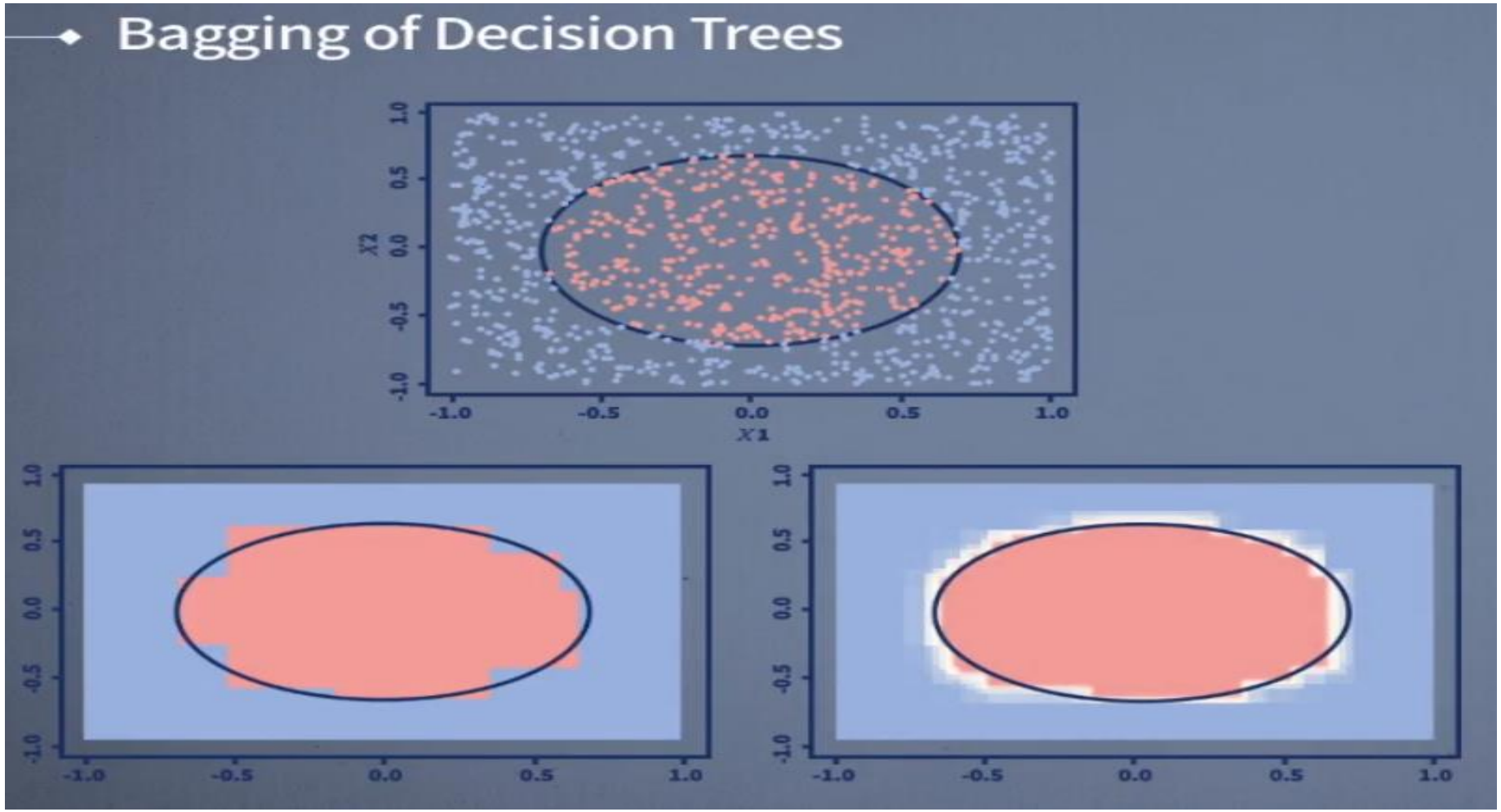
Bagging of Decision Trees

- Training data에서 Training set을 만든다.
  - Ex) data중 100개를 set으로 만들기
- Training set에서 랜덤하게 data를 뽑아낸다.(복원추출)
  - Ex) 복원 추출 : 뽑은 data를 다시 뽑을 수 있음
  - 이를 Bootstrap Samples라고 함
- Bootstrap Samples 마다 Decision Trees를 만든다.
- 모든 결과를 보고 최종 판단을 한다.
- Bagging : bootstrap aggregation

Bagging of Decision Trees



Bagging of Decision Trees



Random Forests

- Decision Tree 여러 개를 통합하자
- Bagging of Decision Tree와 방식은 똑같다.
- 차이 : ‘random’,
  - 입력 변수를 정해 놓고, random으로 선택한 후 선택한 변수에 대해 제일 좋은 하나를 골라서 tree를 만들어 간다.
- 정확도가 떨어질 수 있지만,
  - sample size가 작은 경우 variance를 줄일 수 있다.
  - bagging으로 합치기 때문에 정확도를 cover할 수 있다.
  - 시간을 줄일 수 있다.(모든 feature가 아닌 몇 개의 feature만 선택)

Random Forests

Steps

Let N_{trees} be the number of trees to build

Let m_{try} be the number of features to be selected at each split

몇 개의 Tree를 만들 지, 몇 개의 feature를 선택할 지 먼저 결정한다.

For each of N_{trees} *iterations* 앞서 설명한 과정과 동일

1. Select a new bootstrap sample from training set
2. Grow an un-pruned tree on this bootstrap.
3. At each internal node, randomly select m_{try} features and determine the best split using only these features.
4. Do not perform cost complexity pruning. Save tree as is.

장점

- 많은 dataset에 대하여 높은 accuracy를 보인다.
  - SVM, Neural nets 정도
- Training 시간, model 만드는 시간이 더 적게 걸림
- Parameter의 수 훨씬 더 적다
  - (tree를 몇 개 만들 것인가, 입력을 몇 개 살펴볼 것인가)
- IF-then-else : 사람이 읽을 수 있다.(Interpretable)
- Training 시 overfitting에 대한 저항력이 크다.
- Data에 대한 Preprocessing, outlier 처리를 하지 않아도 된다.
- 입력 변수가 많은데 갖고 있는 data가 별로 없는 경우에 대해서도 정확도가 높다.

Byproduct

- Out-of-bag samples
  - 복원추출 때문에 한 번도 선택되지 않은 data가 나올 수 있음
  - 약 data의 1/3정도가 해당함
  - Out-of bag error 결정 가능
  - Variable Importance 구할 수 있음
  - 각자 bootstrap sample 마다 다름

Out-Of-Bag Error

- Training data에서 bootstrap sample을 만들고, decision tree 100개를 만들었다. Random forest Error를 평가하자.
  - 첫번째 training data를 선택하여, data가 Out-of-Bag에 속해 있는 decision tree를 찾아낸다.
  - 찾은 Decision tree가 10, 15번째라면, 10, 15번째 decision tree에 첫번째 data를 넣어 test를 한다. 그 후 모든 data에 대해 적용하여 error 계산
  - Cross validation, traing & test set을 나눌 필요가 없다.
  - out-of-bag error가 증가/감소함을 통해 stop condition을 정할 수 있다.

Variable Importance

- 어떤 Data가 중요하고, 중요하지 않음을 판단
  - Training data의 첫번째 입력을 randomly shuffling을 한다.
  - Randomly shuffling 한 data를 갖고 out-of bag error을 구하듯 error을 계산한다.
  - 원래의 decision tree와 shuffling해서 나온 tree 값의 차이를 계산한다.
  - ex) x1에 대한 처음 error가 0.1이고, shuffling해서 나온 error가 0.1이다. 이는 x1은 prediction에 중요한 영향을 끼치지 못함을 의미함.