Decision Tree : not Good

Random Forests

Linear Reg
Logistic Reg
Naive Bayesian ⇒ "basic" method (weak model = not Good model)
Decision tree → 성능 일반적 (Not Good)

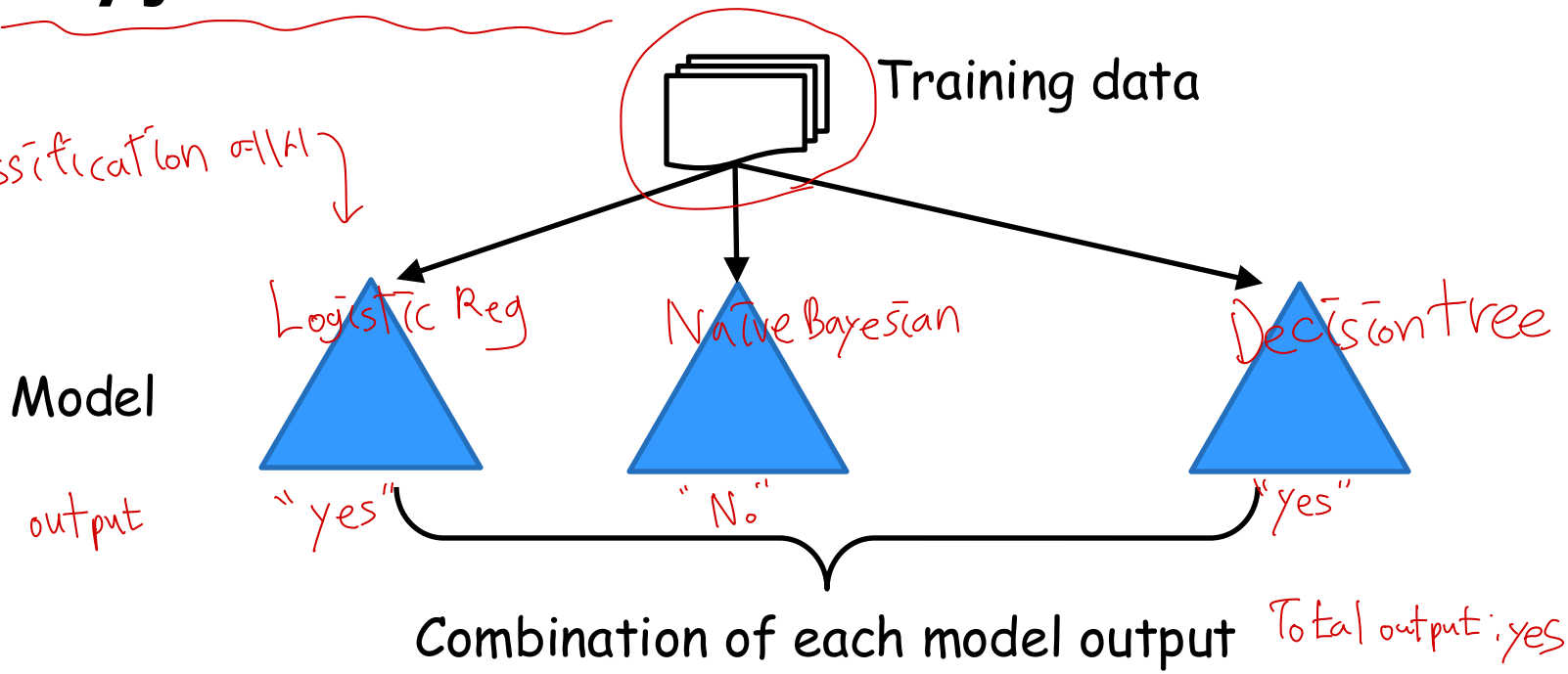
a set of Decision trees
+
Randomness

Random Forest
SVM
NN

strong model

Ensemble

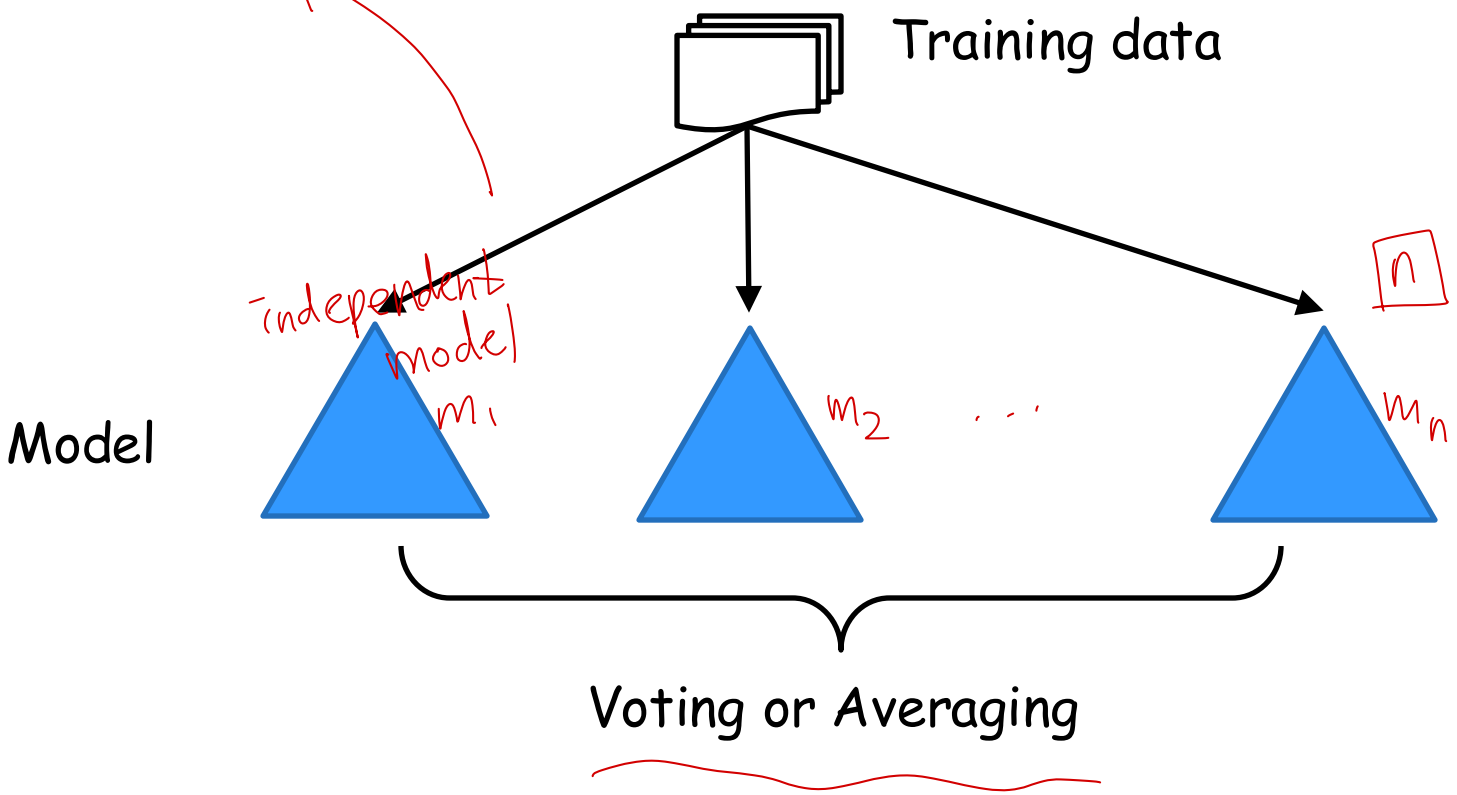
- **Why just one model?**



  - Accuracy of each model should be high enough
  - Correlation of each model should be low enough

Ensemble

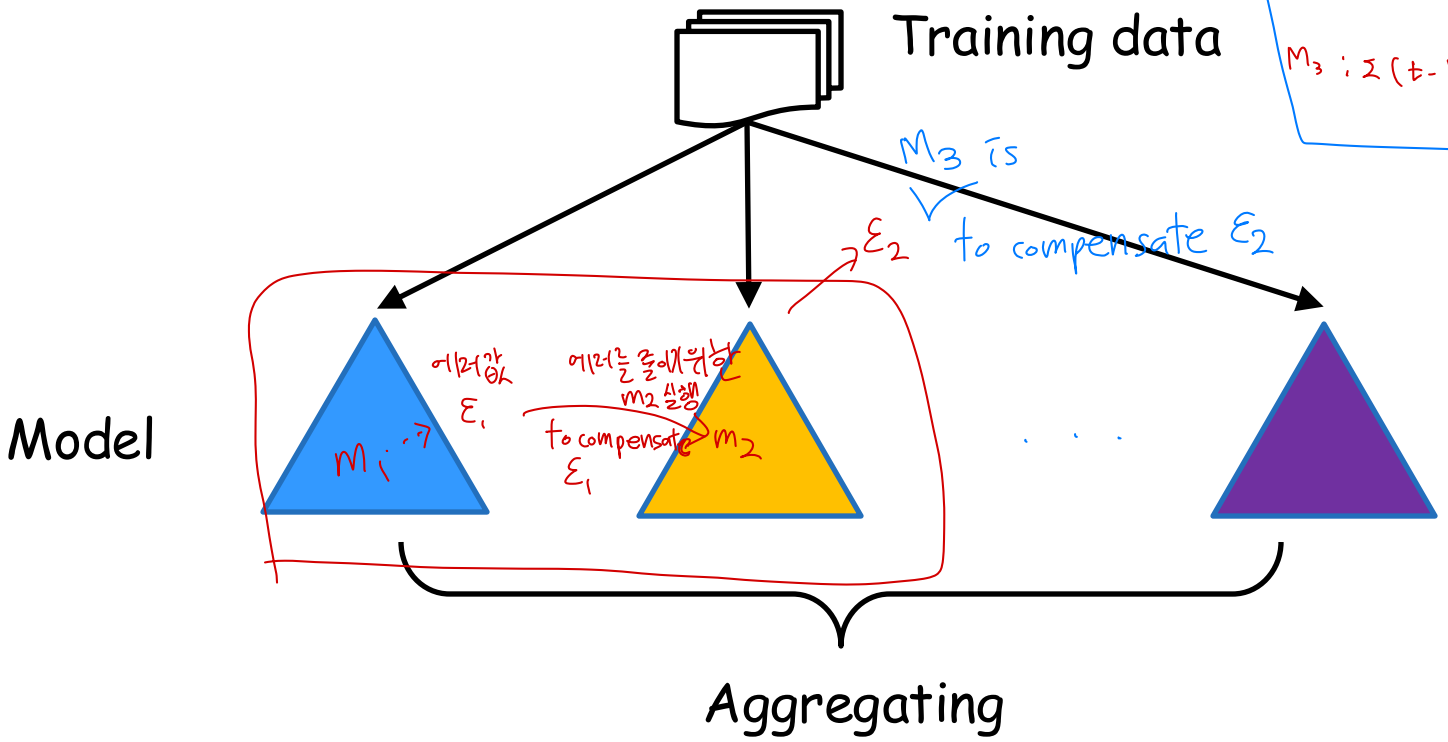
- **Bagging**



  - All models are equivalent

Ensemble

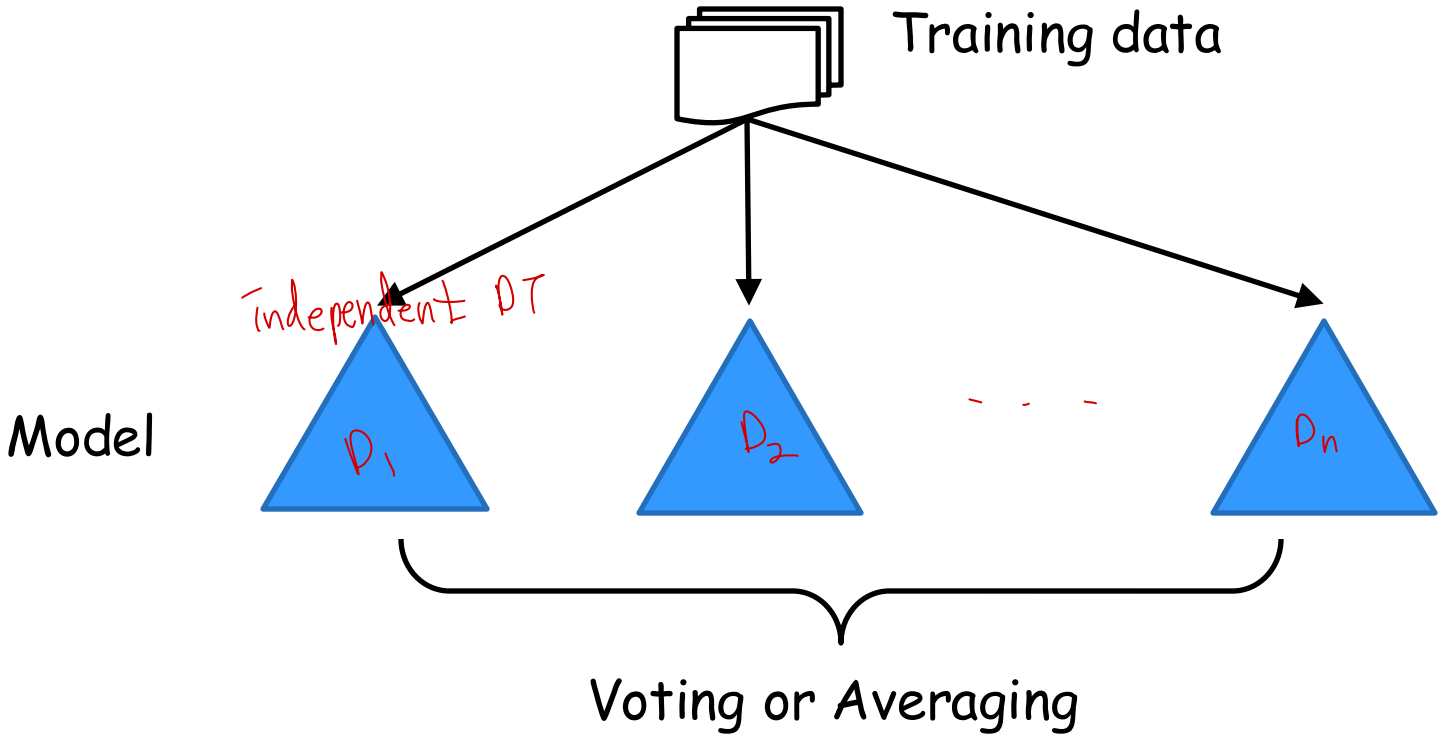
- **Boosting** (Incremental way)



  - Each model compensates the error of the previous models

Bagging of Decision Trees

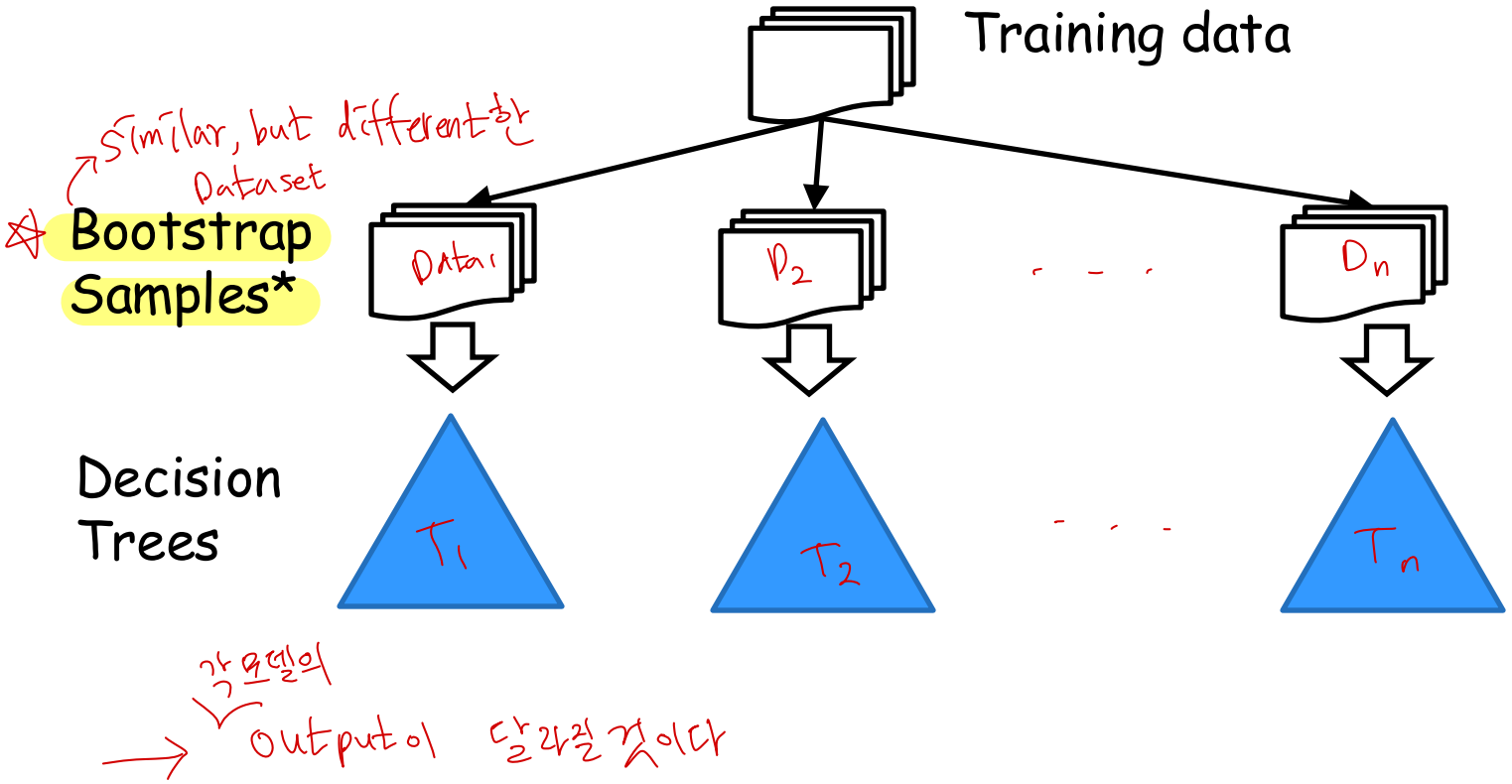
- **Bagging**



- All models must be the same!!

Bagging of Decision Trees

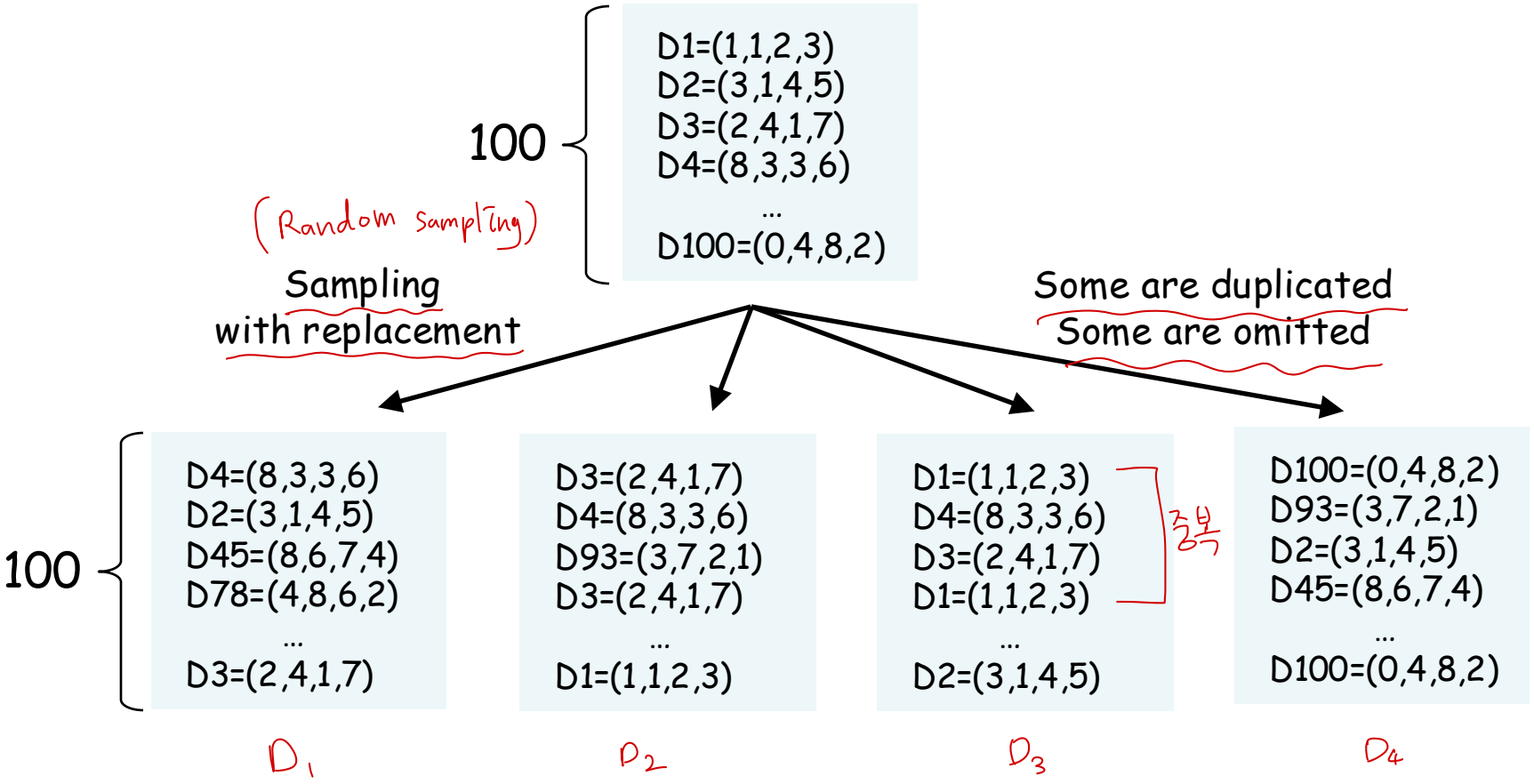
- **Why just one tree?**



*samples equal in size to the original dataset, but selected with replacement

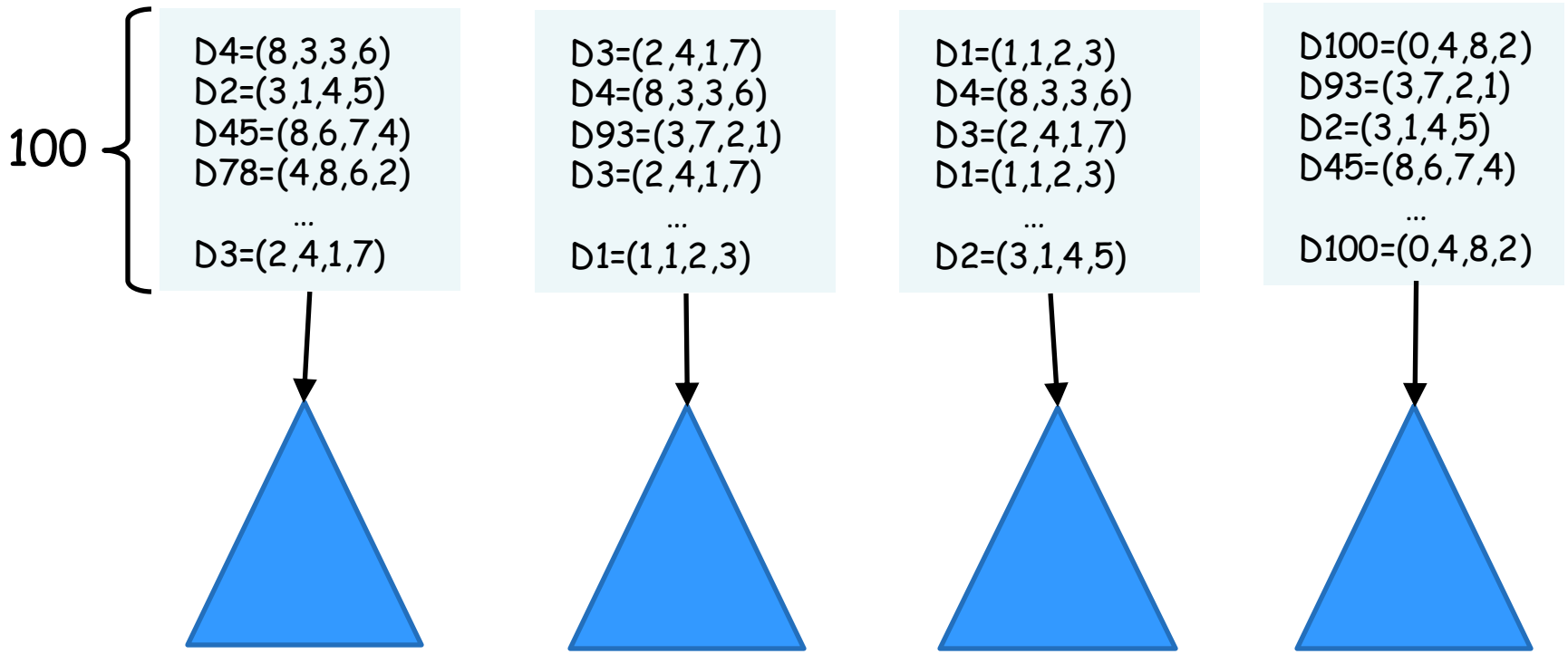
Bagging of Decision Trees

- **Bootstrap Samples**



Bagging of Decision Trees

- **Bootstrap Samples**

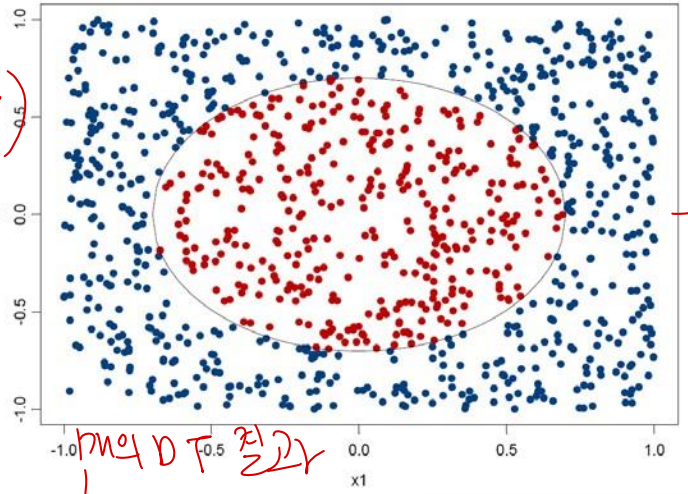


Bagging of Decision Trees

XG"Boost"도 결국은 Tree based 알고리즘이다.

DeepLearning
→ (Lots of Data samples
Time)

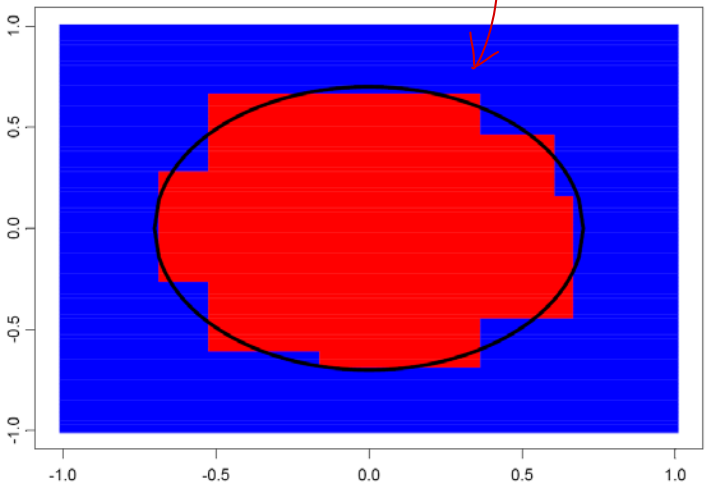
Tree-based ensemble
→ (Good performance
less data
less time & computing)

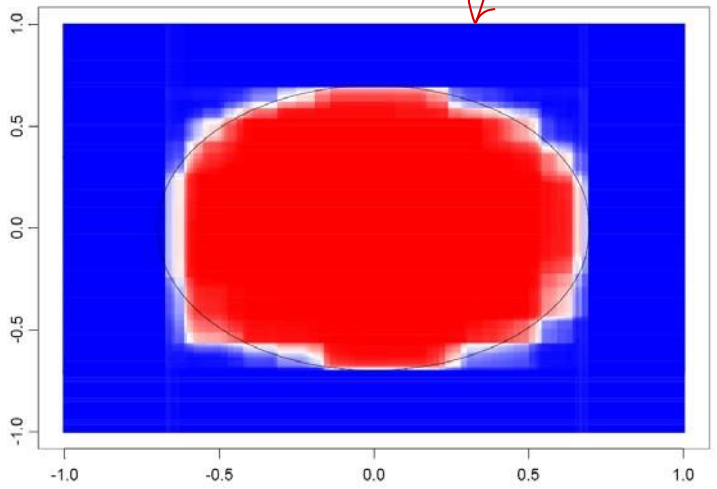


→ Original Dataset

개별 DT 결과

Bagging of DTs





Bagging of Decision Trees

- **Bagging = Bootstrap Aggregation** → 배깅
  - Create multiple models with bootstrap
    - Create new training sets with bootstrap samples
    - Create models with each samples
  - Averaging or major voting of the models

- **Advantage of Bagging**
  - Decreasing test error by lowering prediction variance while leaving bias unchanged

= with weak model과 같은 효과

Random Forests = Bagging of DTs + Randomness

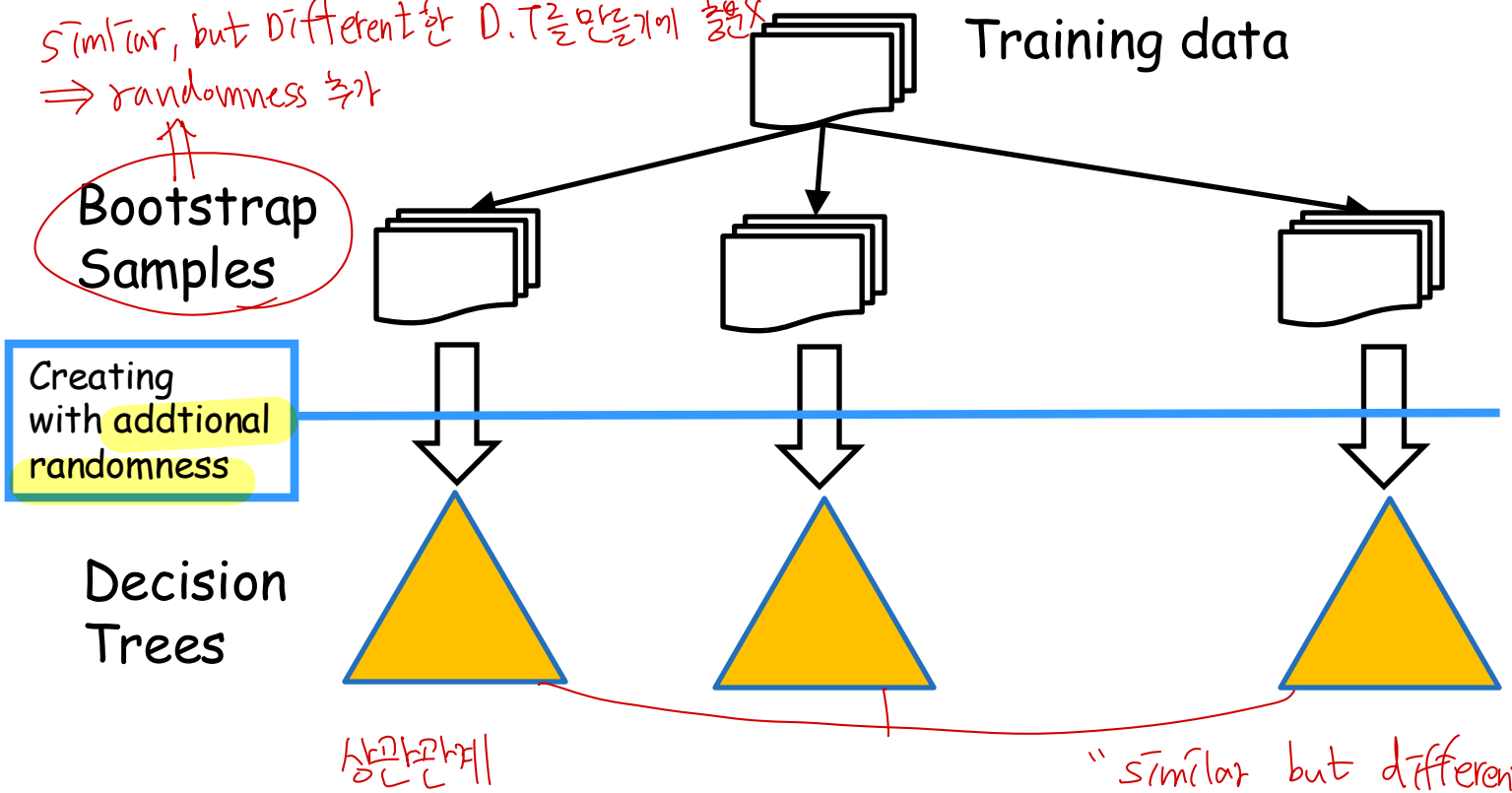
Bootstrap만으로 충분하지 않다

- **Definition**
  - An ensemble method consisting of a bagging of un-pruned decision tree learners
  - with a randomized selection of features at each split.

- **Difference with Bagging**
  - Bagging: selecting the best feature in the full set of feature at each split
  - Random Forests: selecting the best feature in a randomly selection subset of feature at each split (Additional Randomness)

Random Forests

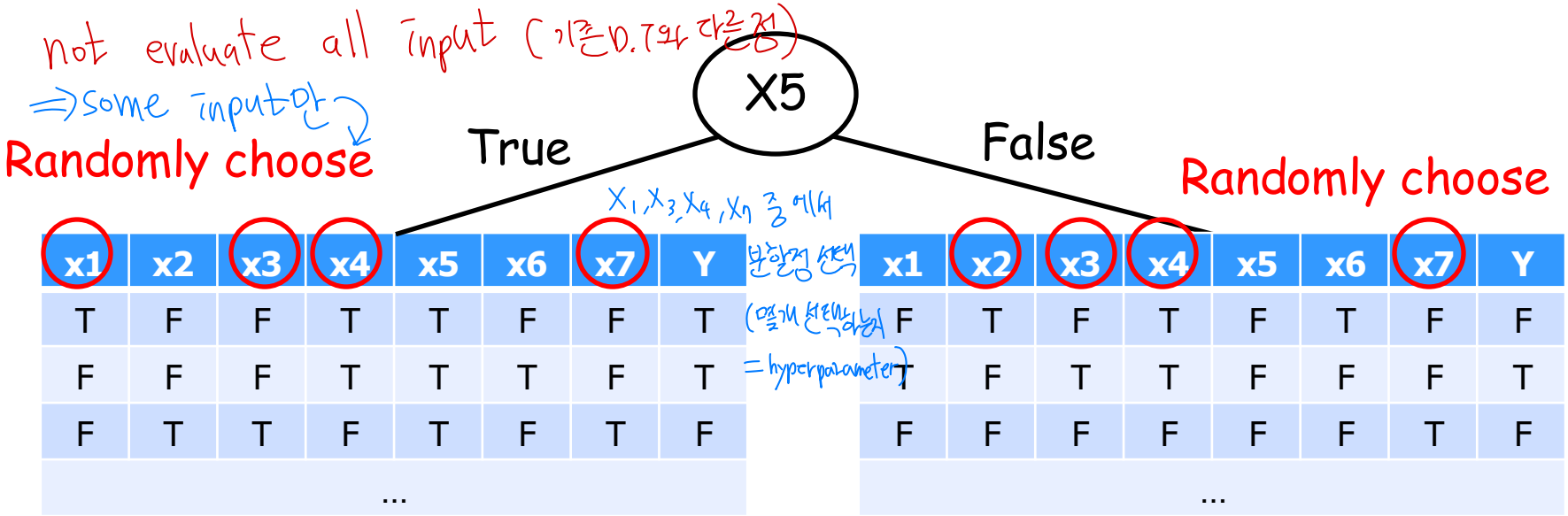
- **Bagging of DT + Additional Randomness**



Corelation between trees are reduced by randomness

Random Forests

- **Additional Randomness**
  - Further reduces variance even on smaller sample set sizes, improving accuracy
  - Subset of features much faster to search than all features



x1	x2	x3	x4	x5	x6	x7	Y
T	F	F	T	T	F	F	T
F	F	F	T	T	T	F	T
F	T	T	F	T	F	T	F
...							

x1	x2	x3	x4	x5	x6	x7	Y
F	T	F	T	F	T	F	F
T	F	T	T	F	F	F	T
F	F	F	F	F	F	T	F
...							

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
=inputs

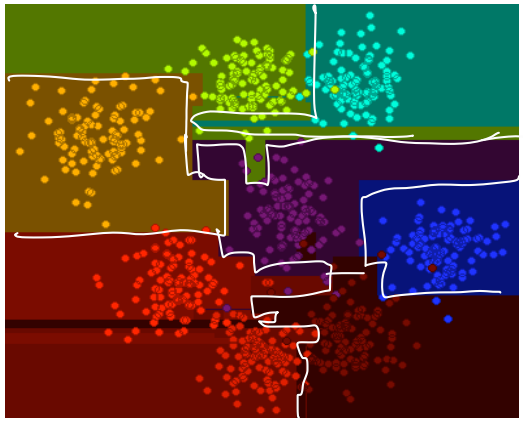
For each of N_{trees} iterations

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.

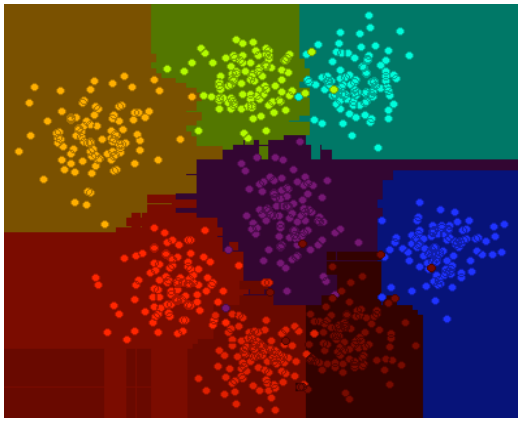
Output overall prediction as the average response (regression) or majority vote (classification) from all individually trained trees

Random Forests

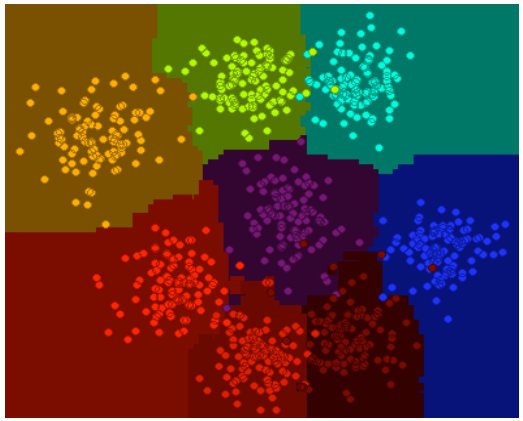
- **Illustration**



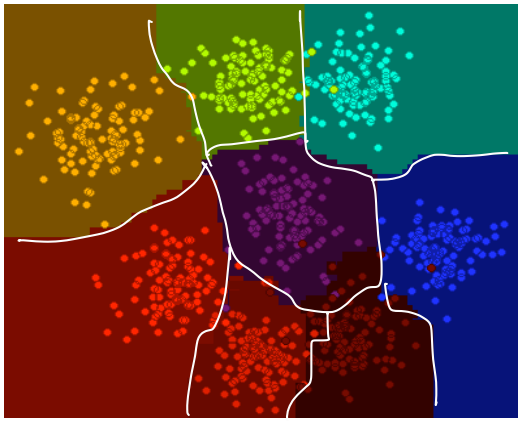
1 Tree



10 Trees



100 Trees

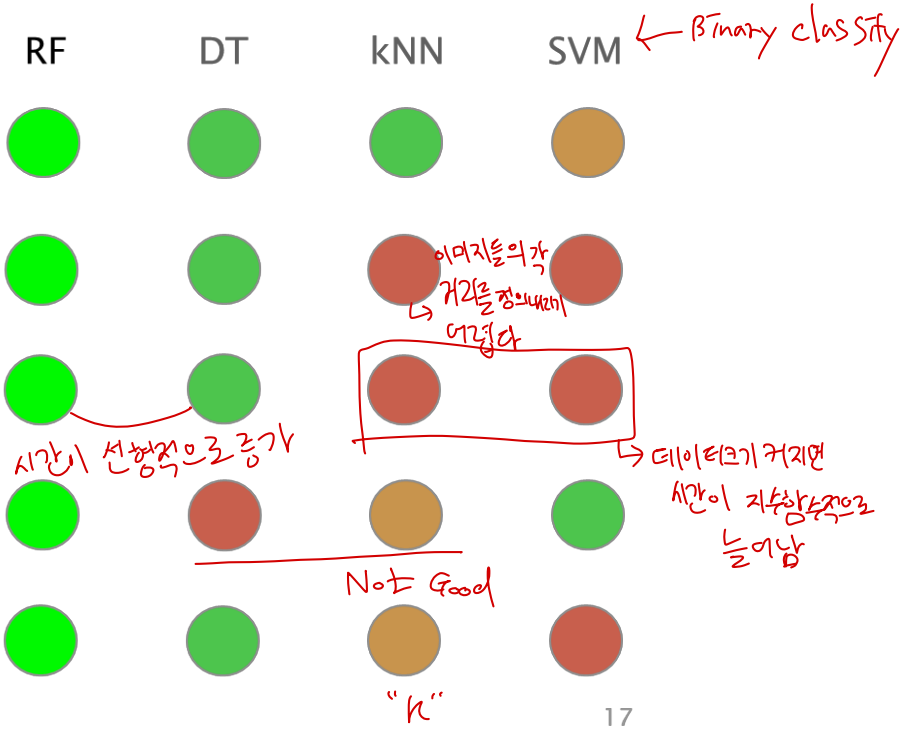


500 Trees

경계가 자연스러움

Comparison

- Intrinsically multiclass
- Handles Apple and Orange features
- Scalability (large learning set)
- Prediction accuracy
- Parameter tuning



Advantages

- **Overall**
  - For many data sets, it produces a highly accurate classifier. (Almost same as SVMs and NNs)
  - It is faster to train and has fewer parameters than SVMs and NNs (good for large databases)
  - It is interpretable

- **Training**
  - It generates an internal unbiased estimate of the generalization error (Cross validation is unnecessary)
  - Resistance to over training

Out-Of-Bag

New concept

R.F 일반적방법 → Train | validation | Test

Out of Bag → Train | Test

= Boot strap | Out of Bag ≠ validation | Test

- **Bootstrap Sampling**

how many samples may not be chosen? ⇒다음장

D1=(1,1,2,3)
D2=(3,1,4,5)
D3=(2,4,1,7)
D4=(8,3,3,6)
...
D100=(0,4,8,2)

What is the prob. that D1 is not include in Bag 1 ?

• Validation Sample은 600개의 D.T의 집합체 ⇒ 전체 R.F 대표, 표준 전체
⇒부분 R.F 대표
• OOB accuracy → D_1 ~ D_{100} 개의 D.T들 (Sub-R.F)

choose D_i as validation Dataset (가정)

$D_1, D_2, D_3, D_4, D_5, D_6$

각각의 $D_1, D_2, \cdots D_{100}$이 존재하는

대신에 accuracy를 측정하여 평균 (≈ Good Estimation of Validation)

= OOB evaluation (InDirect evaluation)

장점 : Training데이터셋의 build, evaluate 동시에 가능하다! Data utilization 좋다!

chosen

100 {
2/3
D4=(8,3,3,6)
D2=(3,1,4,5)
D45=(8,6,7,4)
D78=(4,8,6,2)
...
D3=(2,4,1,7)

D3=(2,4,1,7)
D4=(8,3,3,6)
D93=(3,7,2,1)
D3=(2,4,1,7)
...
D1=(1,1,2,3)

D1=(1,1,2,3)
D4=(8,3,3,6)
D3=(2,4,1,7)
D1=(1,1,2,3)
...
D2=(3,1,4,5)

out of Bag | $D_1D_3D_5$ | $D_2D_4D_6$ | $D_1D_2D_5$ | $D_3D_5D_6$

out of D_1이 있게끔 이 2개의 D.T만 평균냄

만약 D.T가 600개 있다면 D_1이 존재하는 D.T는 약 200개($\frac{1}{3}$)

not chosen ($\frac{1}{3}$) 33% | 또다른 33% | ... | 33% → Out-of-Bag

out of Bag

Out-Of-Bag

- **Bootstrap sample set from learning set D**
  - Remaining samples called out-of-bag samples (OOB)
  - About 33% of original data are not selected in bootstrap

선택되지 않을 확률

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{n} = e^{-1} = \frac{1}{2.718\cdots} \approx 33\%$$

= not chosen의 확률 (데이터 1개)

Data (n = Data 개수)

D1=(1,1,2,3)
D2=(3,1,4,5)
D3=(2,4,1,7)
D4=(8,3,3,6)
...
D100=(0,4,8,2)

Sampling with replacement

Some are duplicated
Some are omitted

D4=(8,3,3,6)
D2=(3,1,4,5)
D45=(8,6,7,4)
D78=(4,8,6,2)
...
D3=(2,4,1,7)

Bootstrap sample