Machine Learning Ensemble Boosting

전창욱

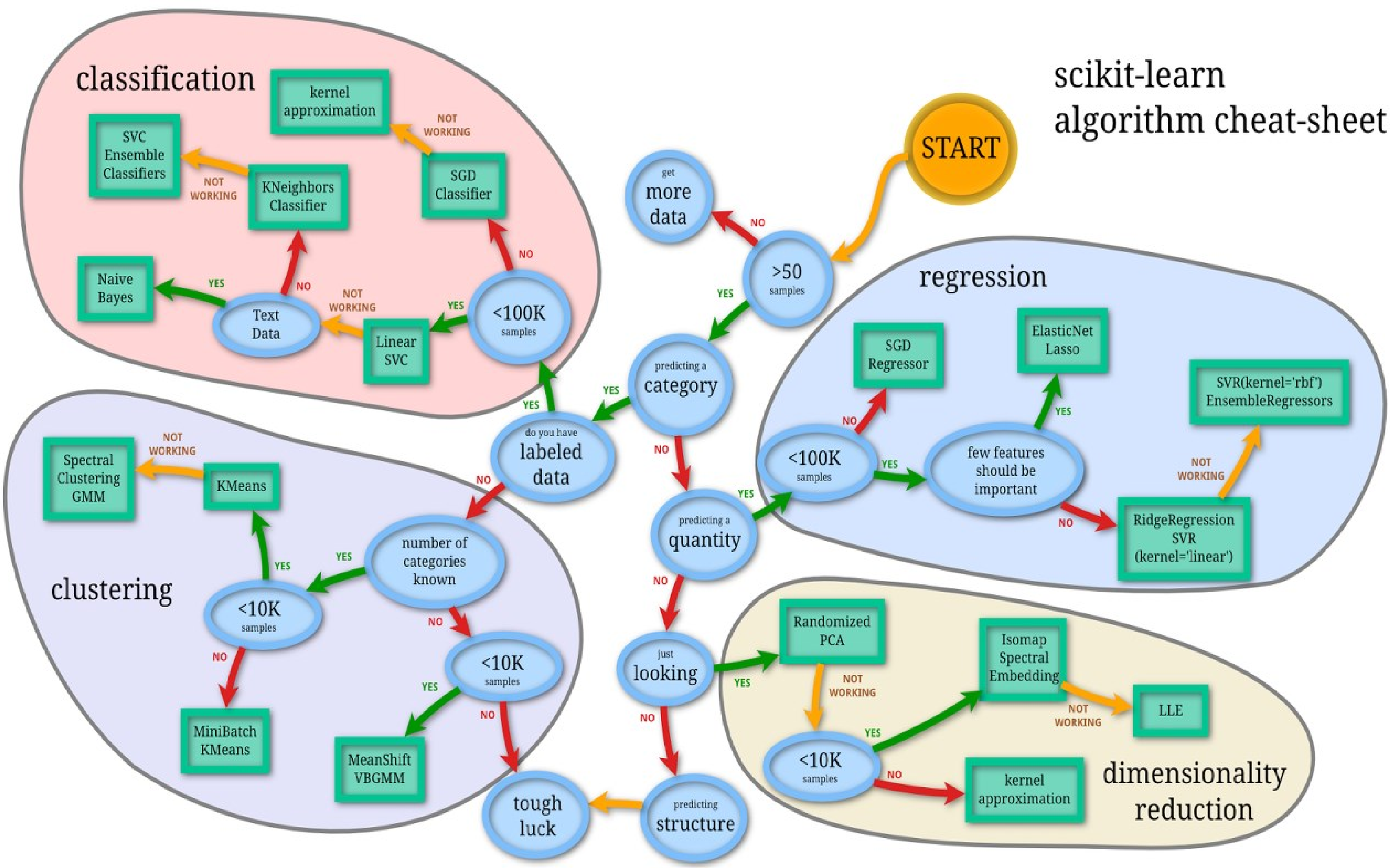
2018.04.24

목차

1. WHY
2. Ensemble
3. Bagging
4. Boosting
5. Stacking
6. Adaptive Boosting
7. GBM
8. XG Boost
9. Light GBM

XGBoost

XGBoost


scikit-learn
algorithm cheat-sheet
START
>50 samples
get more data
predicting a category
do you have labeled data
predicting a quantity
just looking
predicting structure
tough luck
classification
kernel approximation
SVC Ensemble Classifiers
KNeighbors Classifier
SGD Classifier
Naive Bayes
Text Data
Linear SVC
<100K samples
clustering
Spectral Clustering GMM
KMeans
number of categories known
<10K samples
MiniBatch KMeans
MeanShift VBGMM
regression
SGD Regressor
ElasticNet Lasso
SVR(kernel='rbf') EnsembleRegressors
few features should be important
RidgeRegression SVR (kernel='linear')
dimensionality reduction
Randomized PCA
Isomap Spectral Embedding
LLE
YES
NO
NOT WORKING

XGBoost

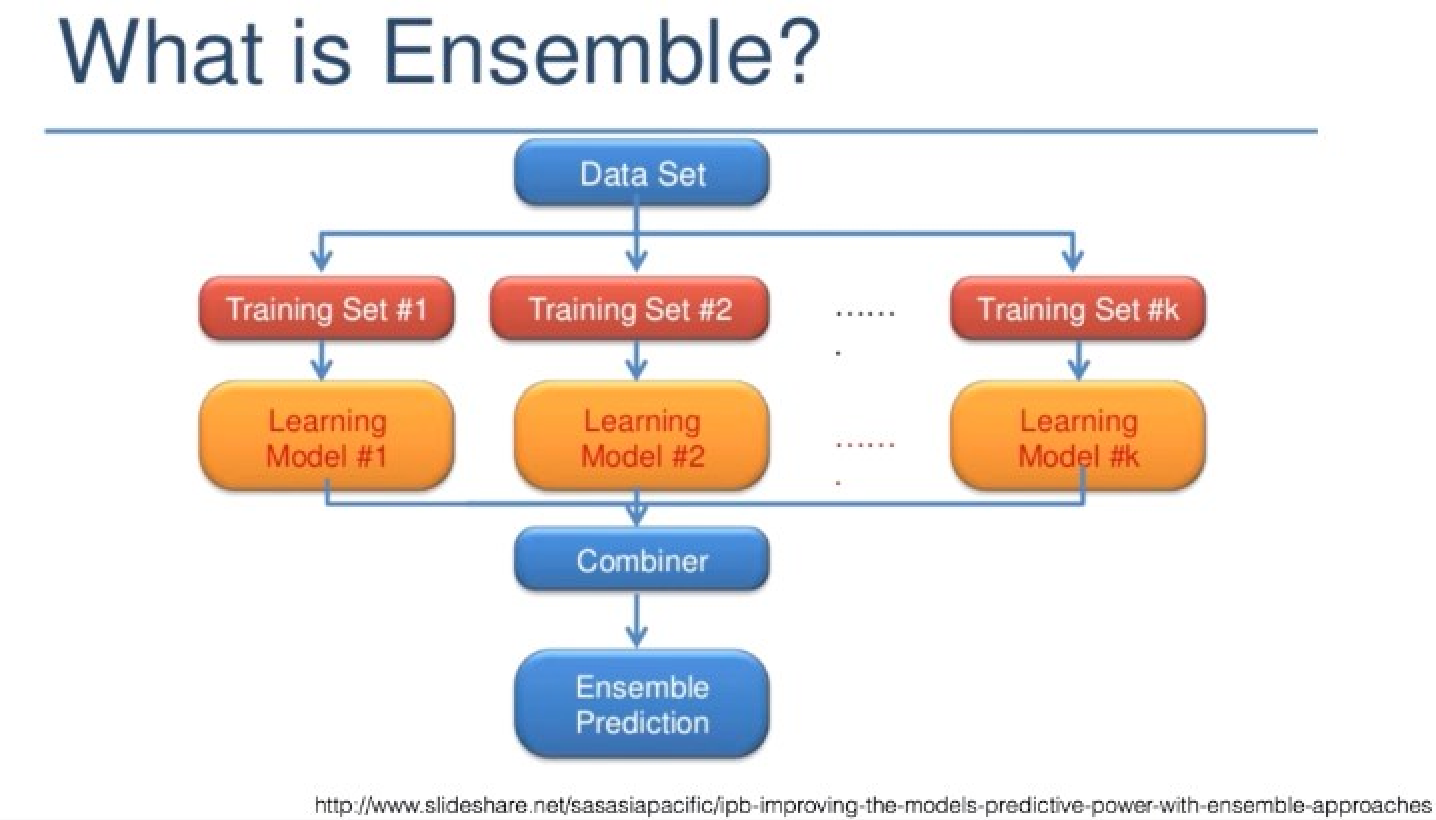
What is XGBoost?

XGBoost는 eXtream Gradient **Boosting**의 약자

Gradient Boosting Algorithm

최근 Kaggle User에게 큰 인기

Ensemble



동일한 학습 Algorithm을 사용 여러 Model 학습

Weak learner를 결합하면, Single learner보다 나은 성능

Ensemble

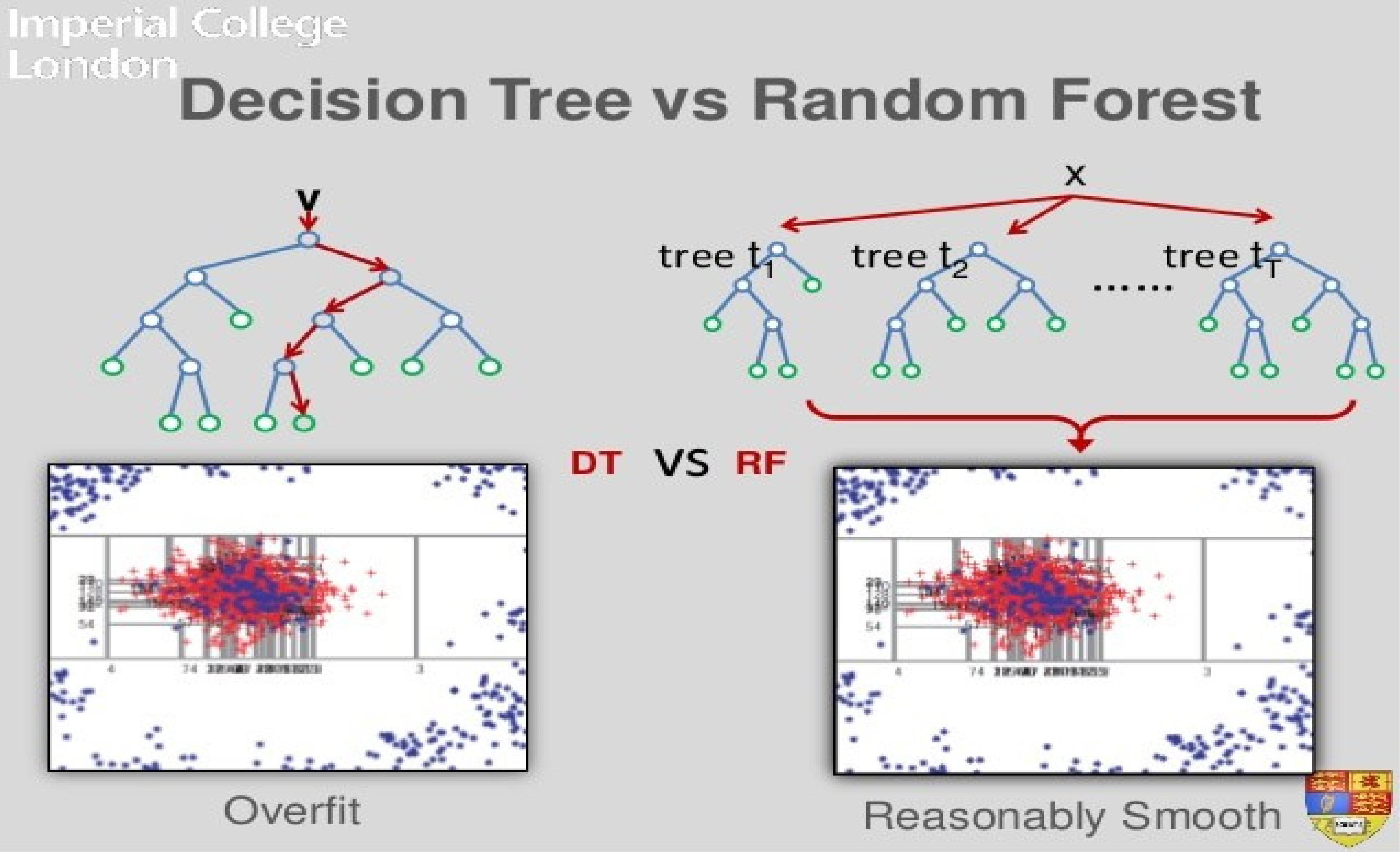
3.2.4.3.1. sklearn.ensemble.RandomForestClassifier

RandomForestClassifier Ensemble 이네

ML에서는 아주 쉽게 Ensemble을 지원하네

컴퓨터가 많고 데이터가 무조건 많아야 할 수 있는게 아니네

Random Forest



Bagging and Boosting and Stacking

서로 다른 모델을 결합하여 새로운 모델을 만드는 것

동일한 학습 알고리즘을 사용하는 방법을 Ensemble

	Bagging	Boosting	Stacking
Partitioning of the data into subsets	Random	Giving mis-classified samples higher preference	Various
Goal to achieve	Minimize variance	Increase predictive force	Both
Methods where this is used	Random subspace	Gradient descent	Blending
Function to combine single models	(Weighted) average	Weighted majority vote	Logistic regression

Bagging

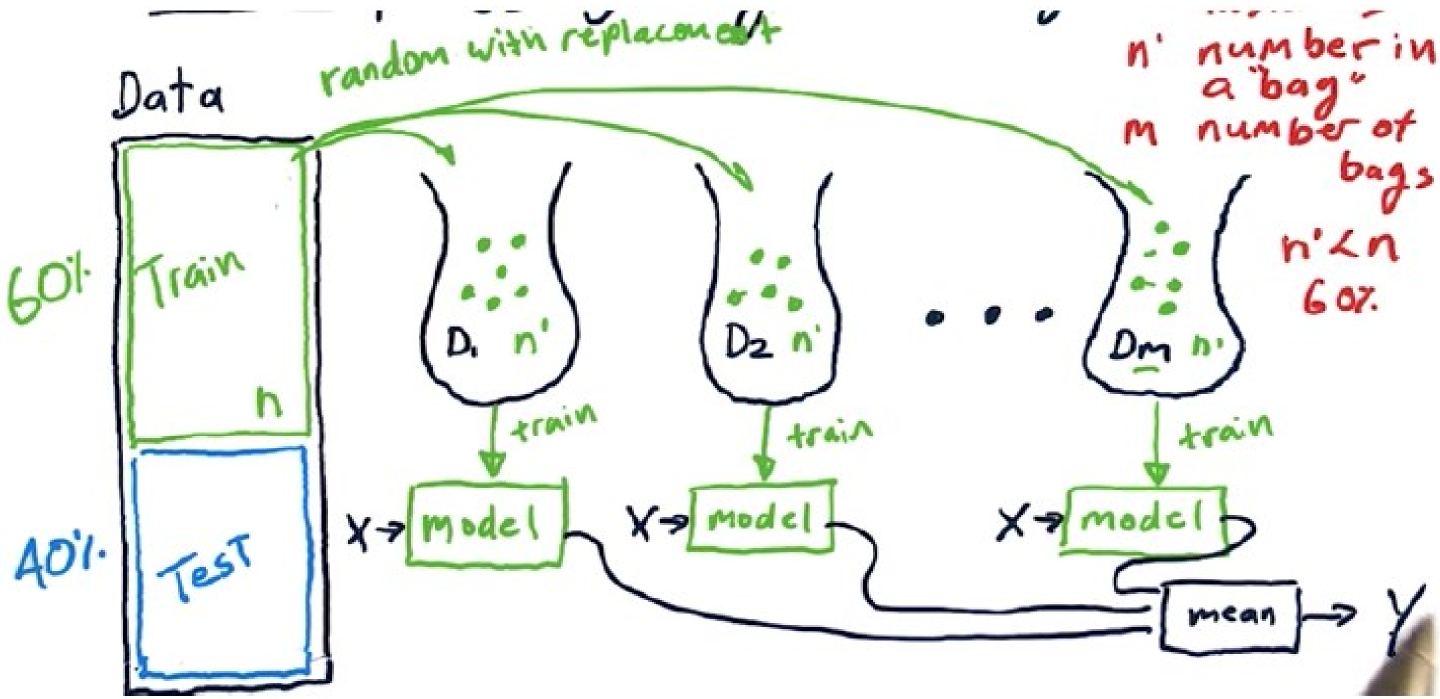
학습 데이터를 랜덤으로 Sampling 하여 여러 개의 Bag으로 분할하고,

각 Bag별로 모델을 학습한 후, 각 결과를 합하여 최종 결과를 도출

n: 전체 학습 data 수

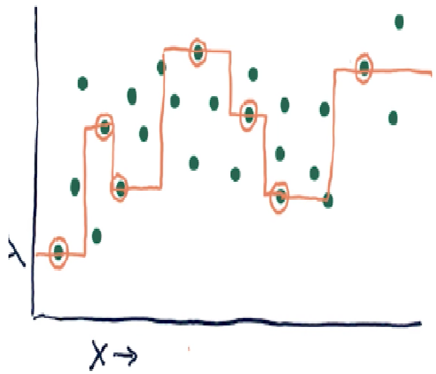
n‘: bag에 포함된 data 수, 전체 data 중 sampling data

m: bag의 개수, 학습할 모델별로 sampling data set

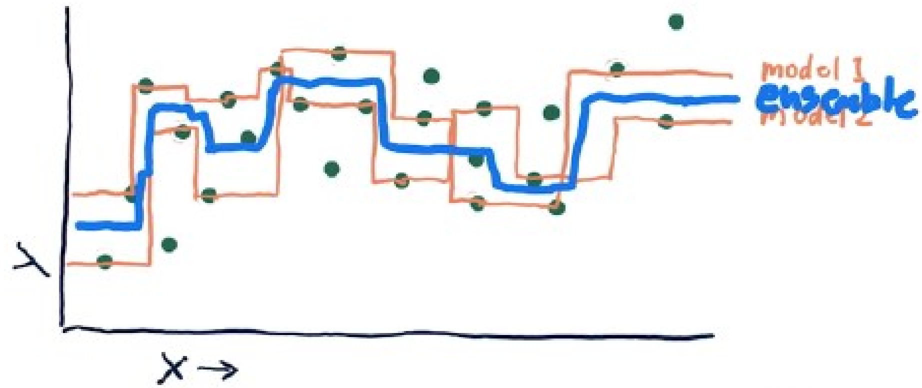


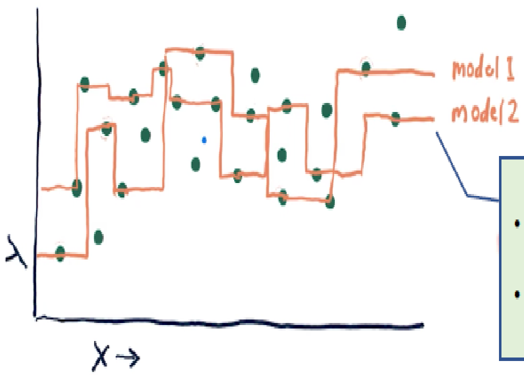
Bagging (Regression)

Low Bias



- 샘플링 된 데이터는 잘 예측하는 모델
- 예측값이 정답 근처에 있음





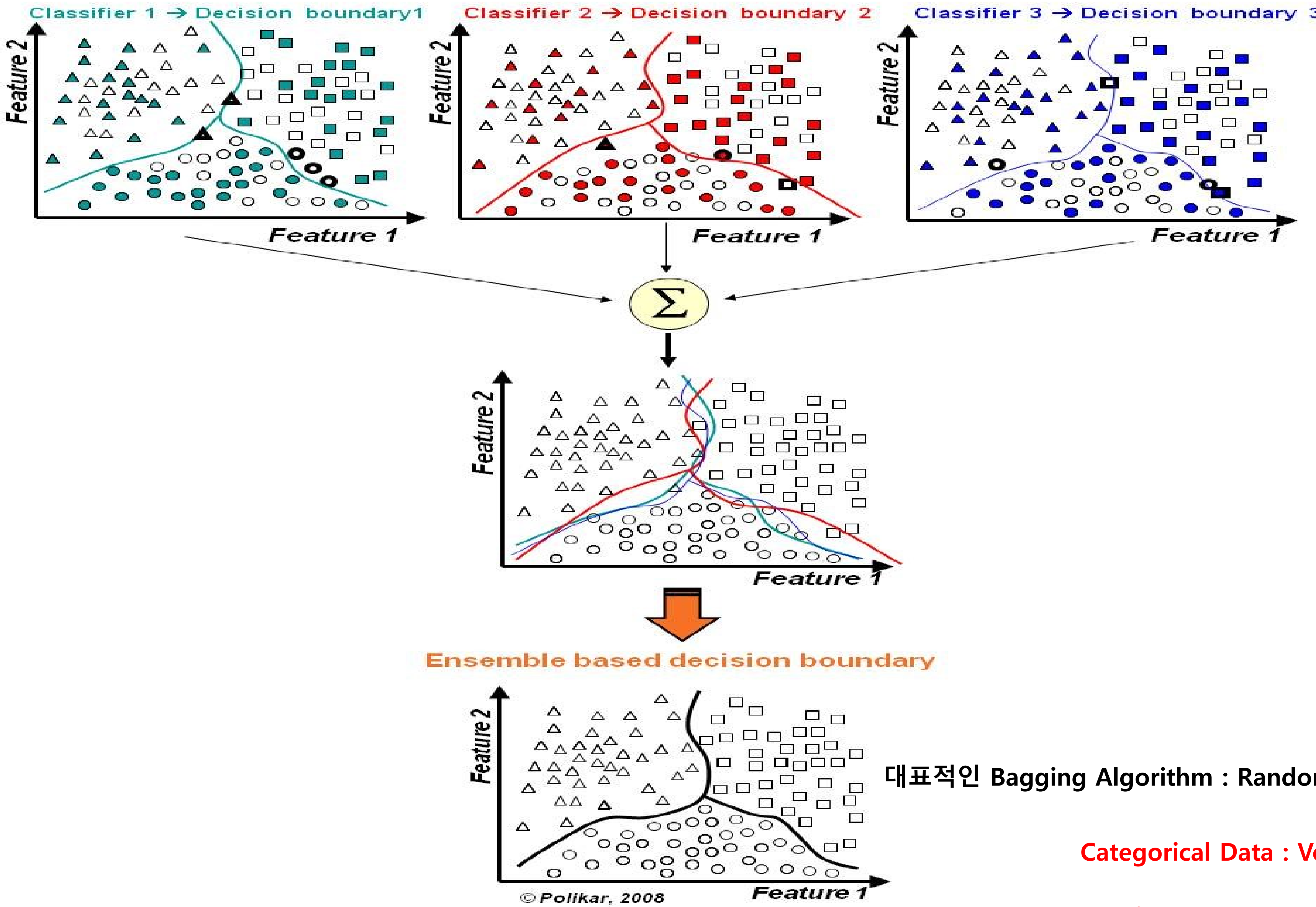
- model 1과 model2가 예측한 값이 서로 많이 다름
- High variance

High Variance

Low Variance

http://scott.fortmann-ro

Bagging



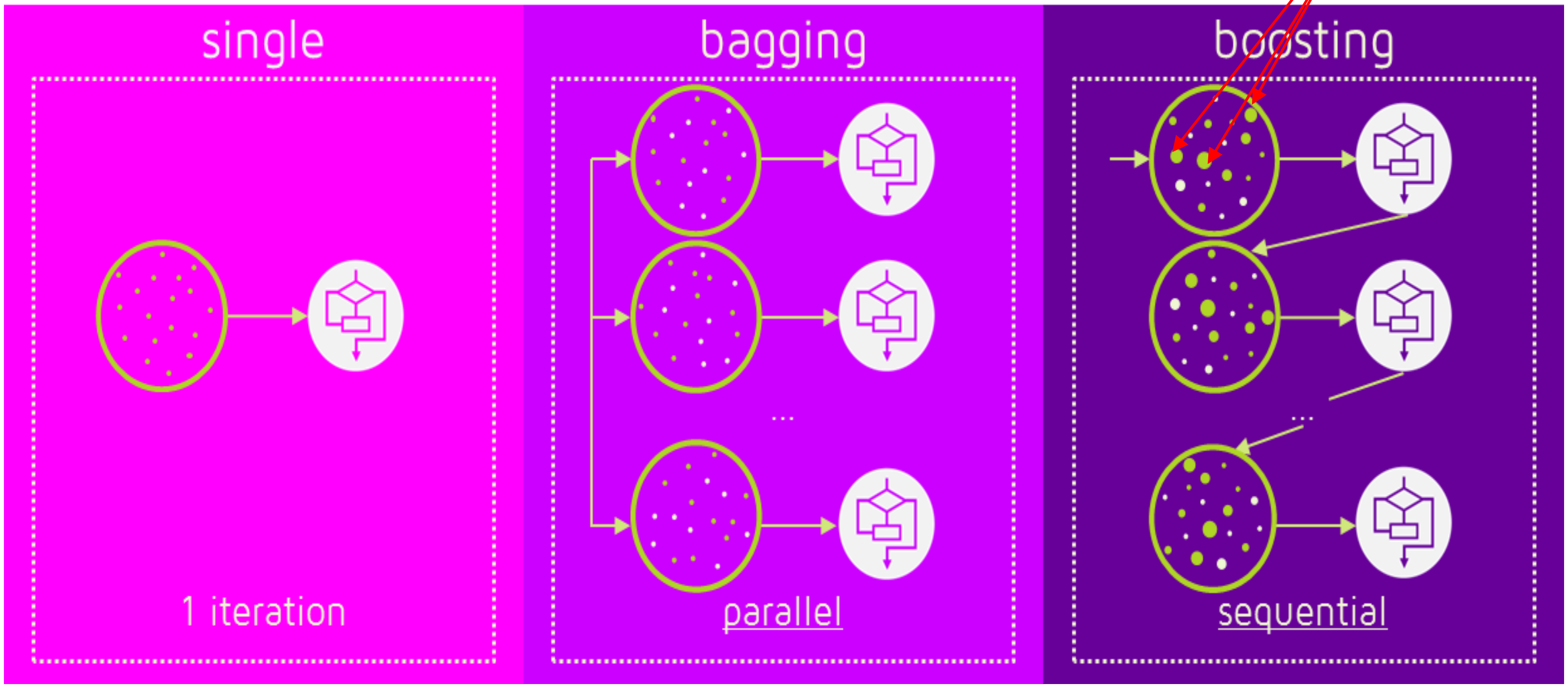
대표적인 Bagging Algorithm : Random Forest

Categorical Data : Voting

Continuous Data : Average

Boosting

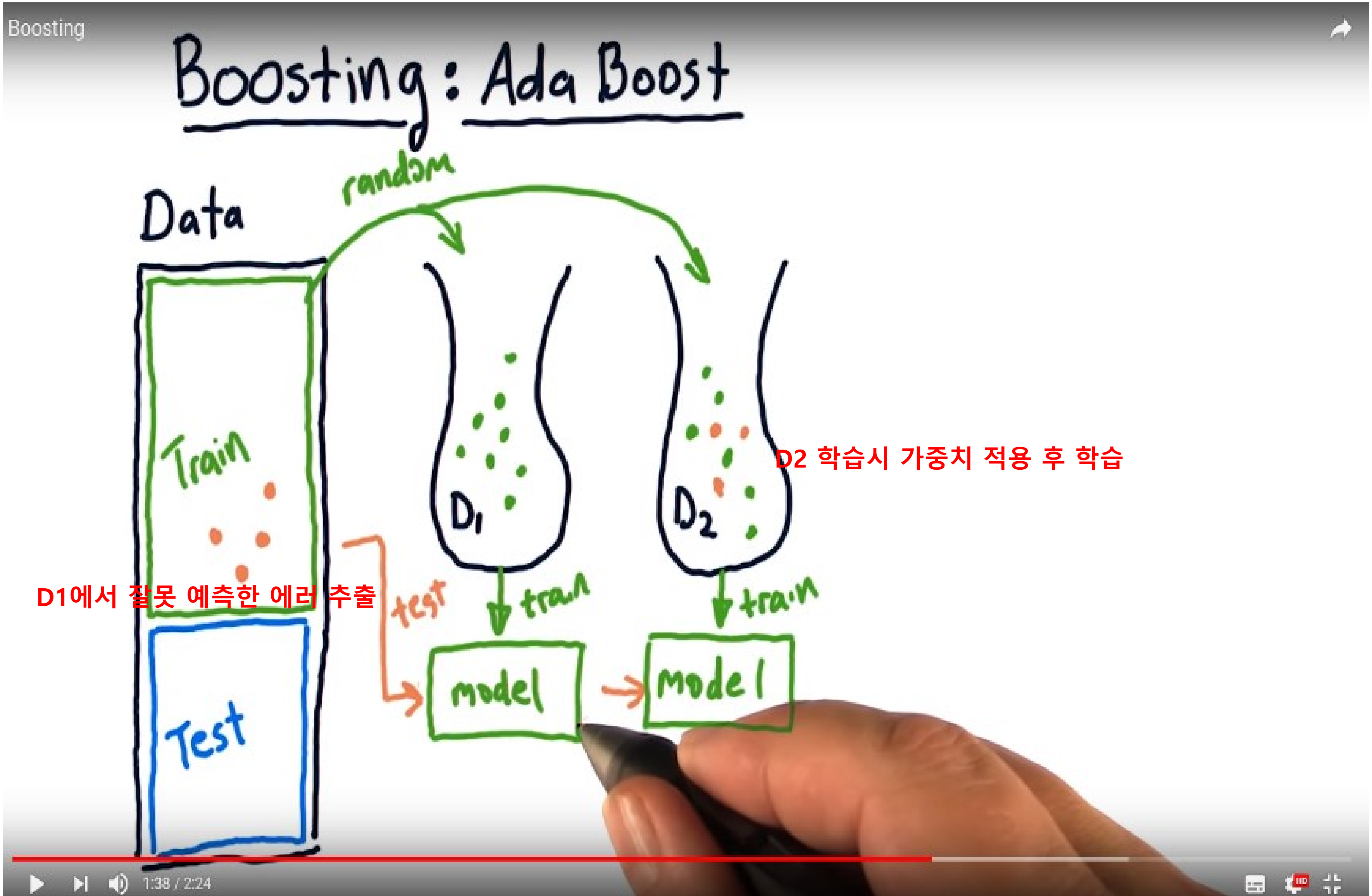
가중치를 부여



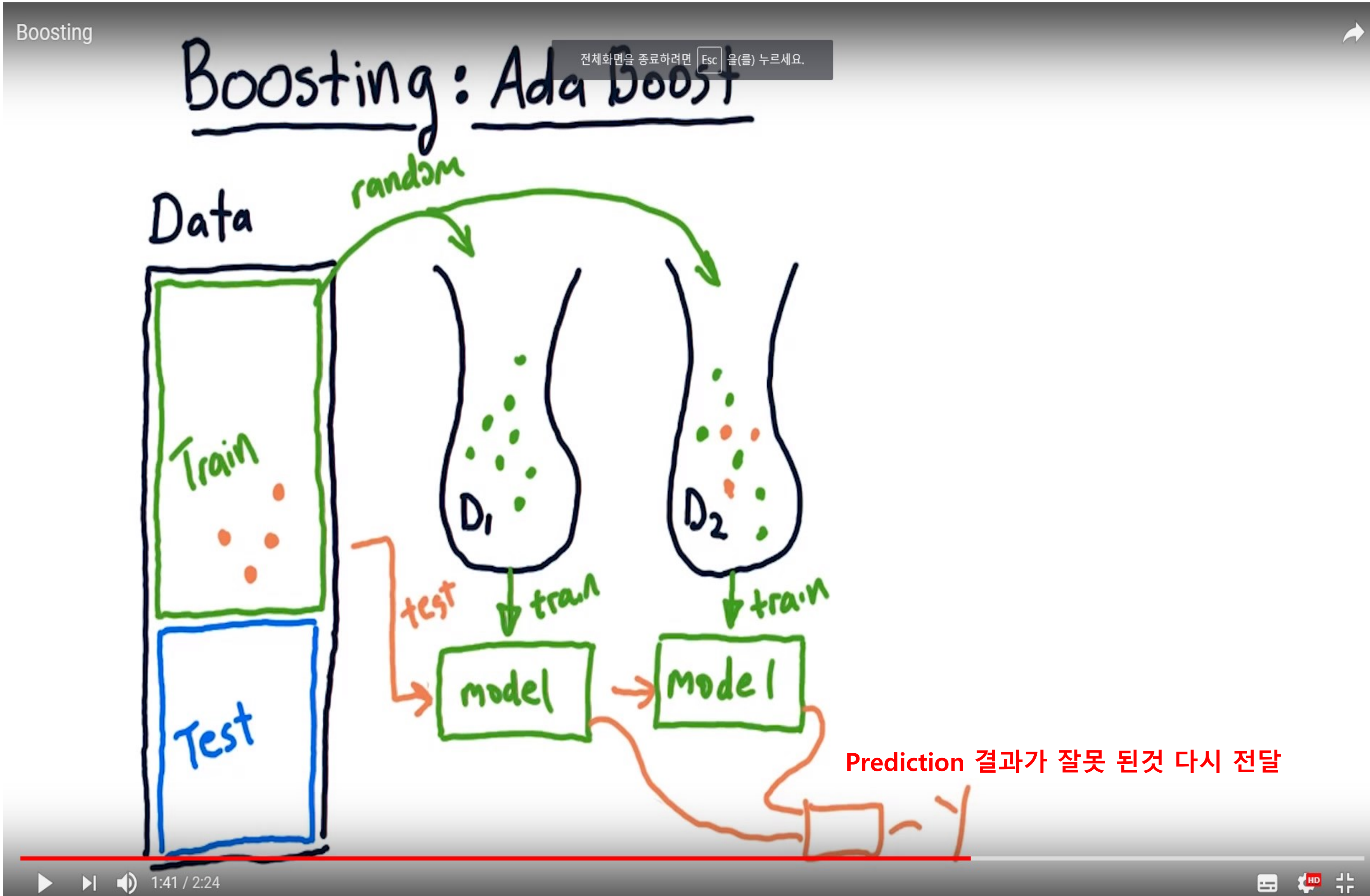
정확도

Outlier

Boosting

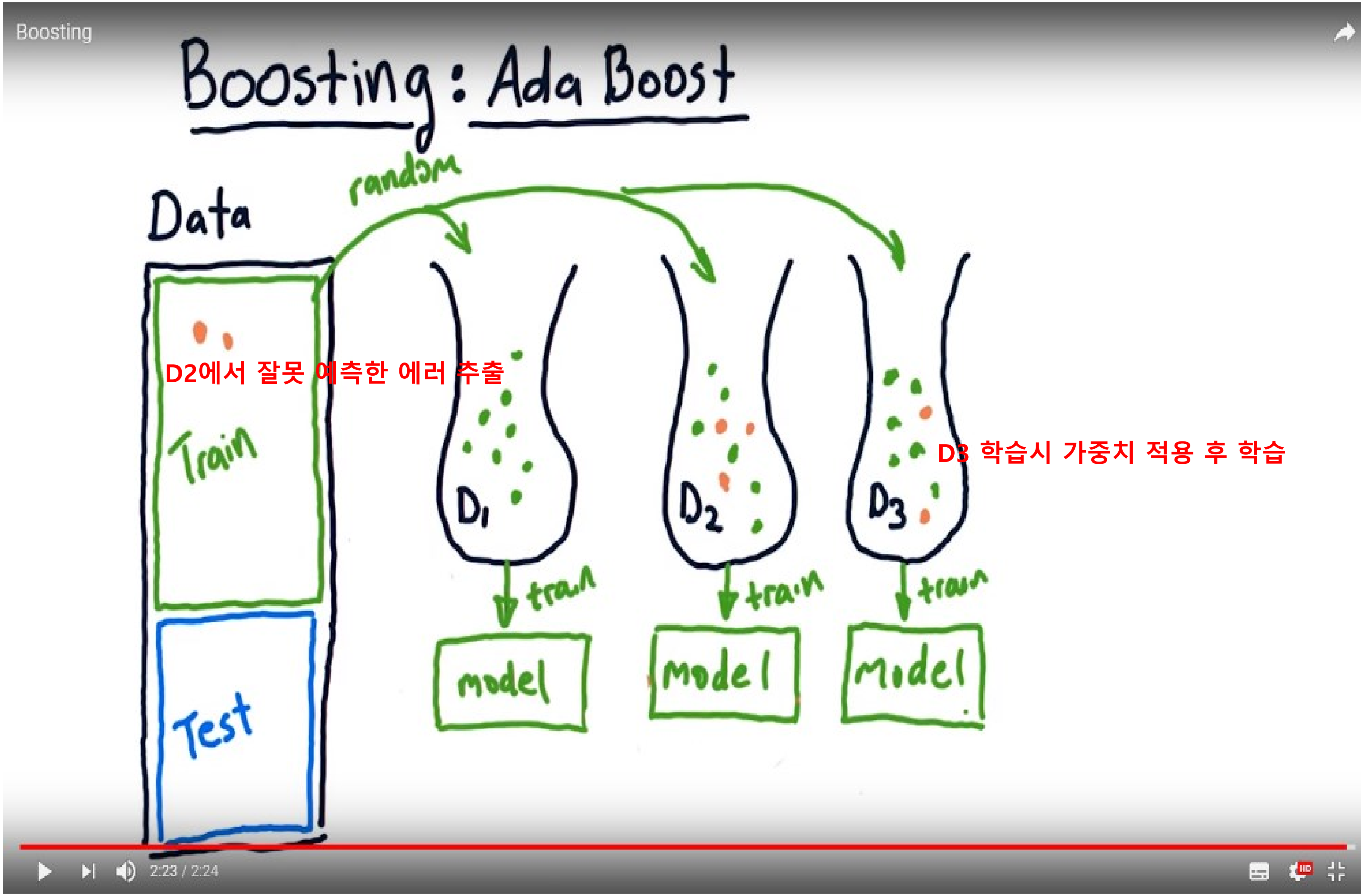

Boosting
Boosting : Ada Boost
random
Data
Train
Test
D1
D2
test
train
train
model
model
1:38 / 2:24
D2 학습시 가중치 적용 후 학습
D1에서 잘못 예측한 에러 추출

Boosting



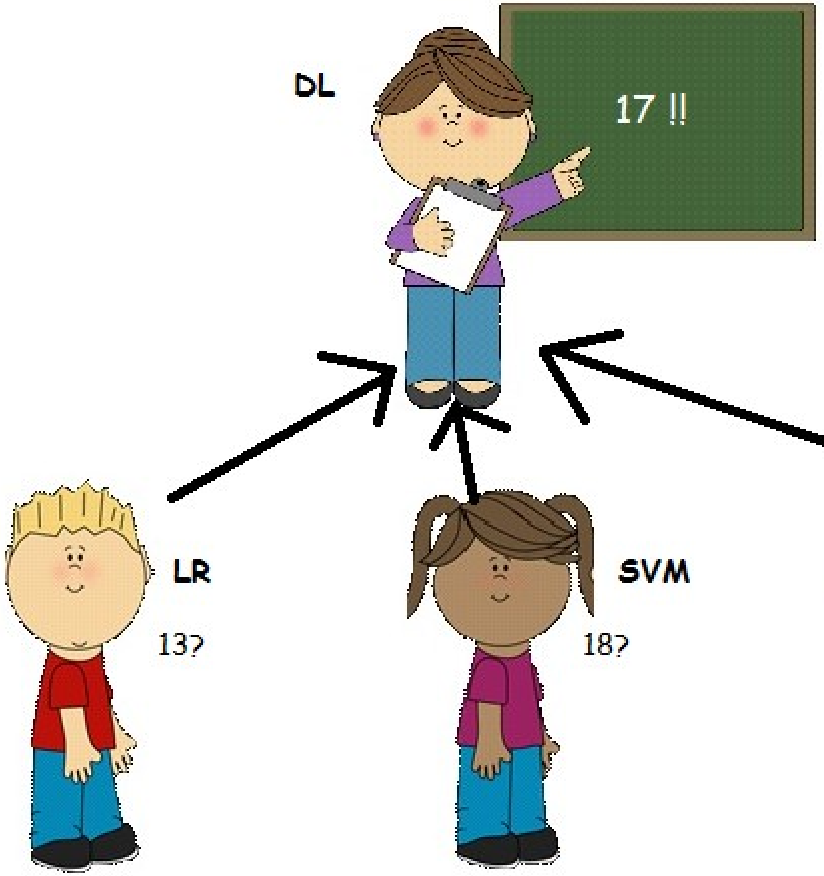
Prediction 결과가 잘못 된것 다시 전달

Boosting


Boosting
Boosting : Ada Boost
Data
random
Train
Test
D2에서 잘못 예측한 에러 추출
D3 학습시 가중치 적용 후 학습
train
train
train
model
Model
Model
2:23 / 2:24

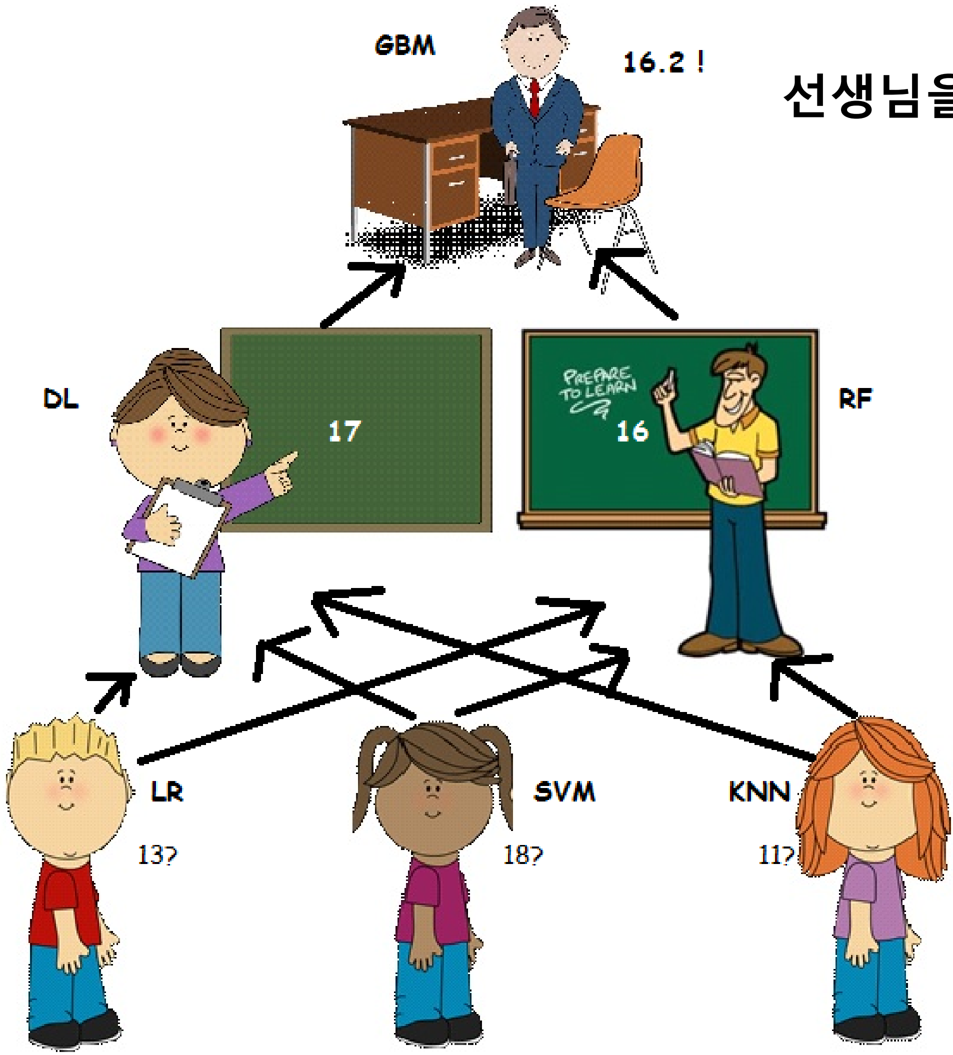
Stacking

Meta Modeling “Two heads are better than one”



학생을 잘 파악하고 있음

Stacking



선생님을 잘 파악하고 있음

StarkNet https://github.com/kaz-Anova/StackNet

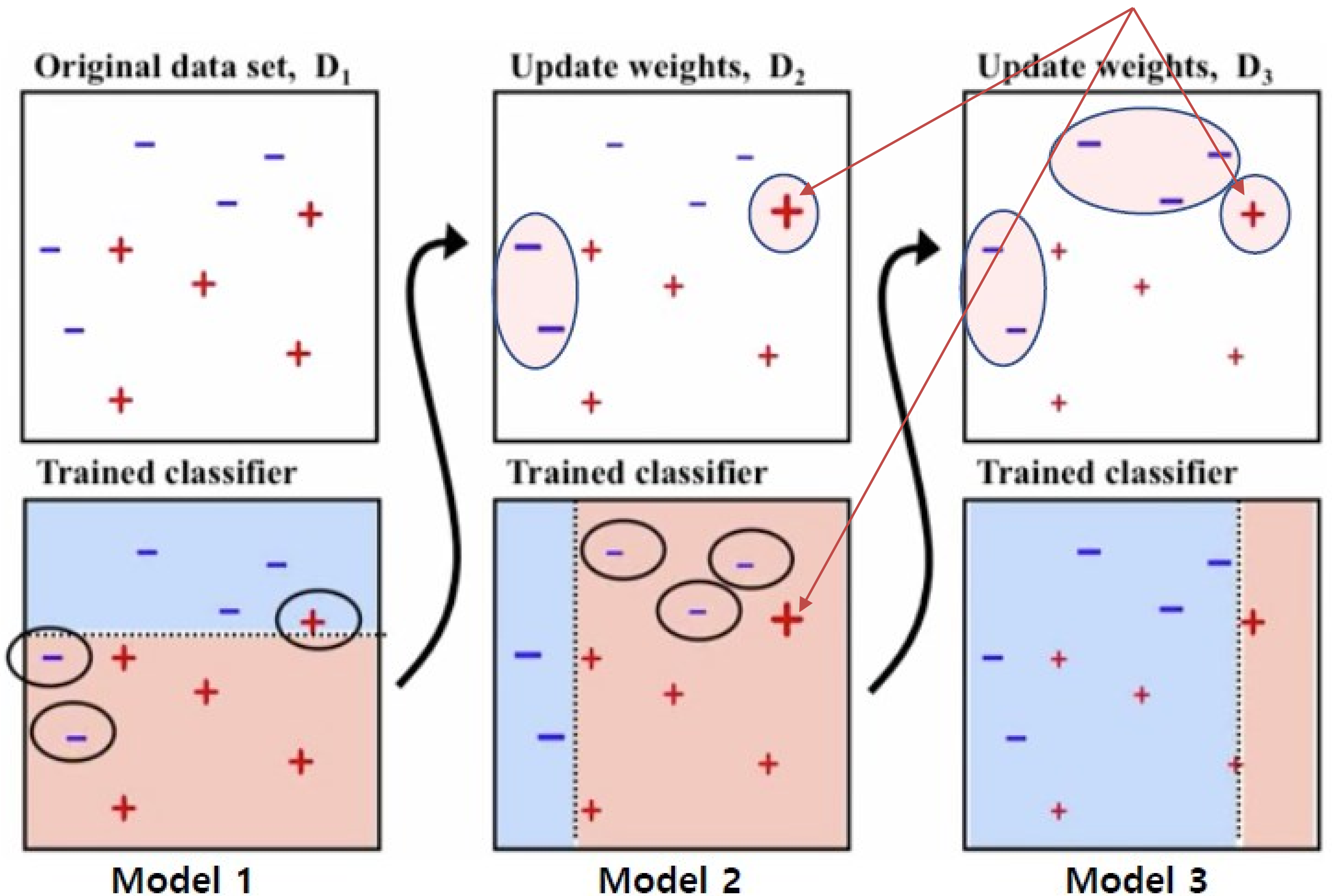
Boosting

Boosting Algorithm

Algorithm	특징	
AdaBoost	다수결을 통한 정답 분류 및 오답에 가중치 부여	
GBM	Loss Function의 Gradient를 통해 오답에 가중치 부여	
Xgboost	GBM 대비 성능 향상 시스템 자원 효율적 활용(CPU, Mem) Kaggle을 통한 성능 검증(많은 상위 랭커가 사용)	2014년 공개
Light GBM	Xgboost 대비 성능 향상 및 자원소모 최소화 Xgboost가 처리하지 못하는 대용량 데이터 학습 가능 Approximates the split을 통한 성능 향상	2016년 공개

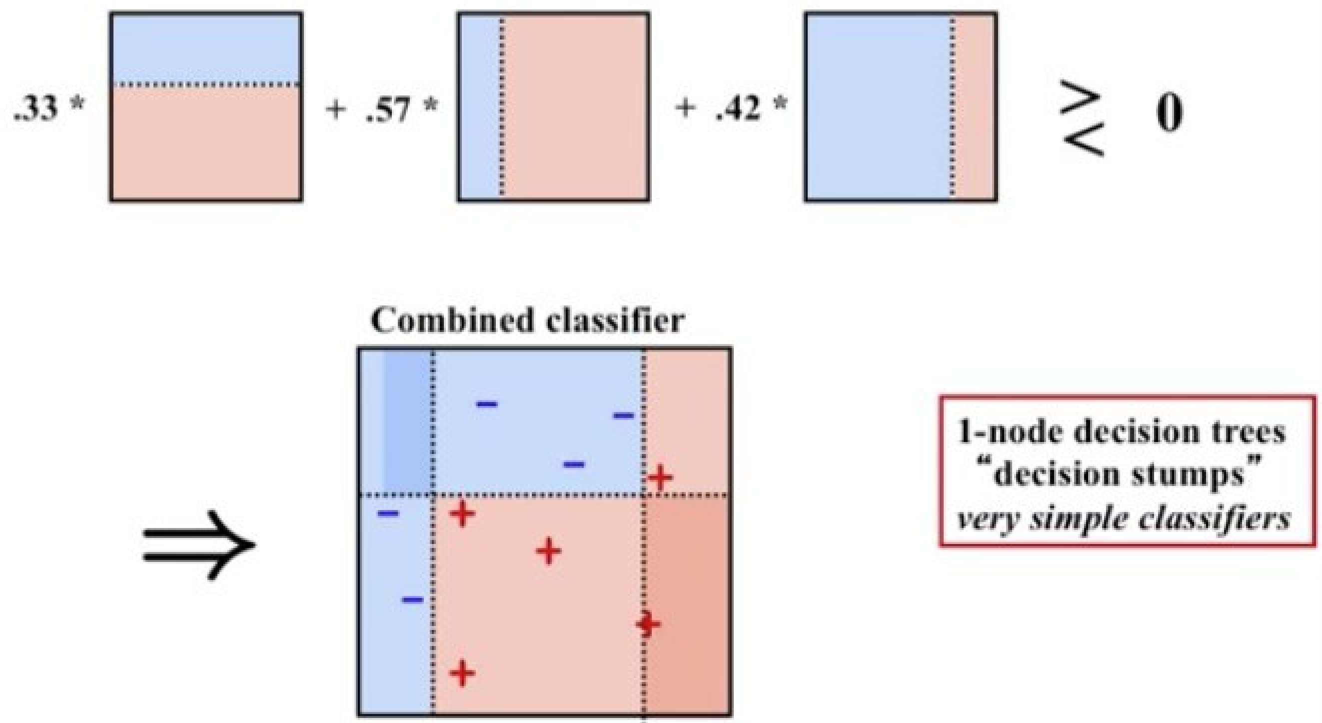
AdaBoost (Adaptive Boosting)

분류를 못하면 가중치가 계속 남음



AdaBoost (Adaptive Boosting)

$$J(\theta) = \sum_i w_i J_i(\theta, x^{(i)})$$ Cost Function : 가중치(W)를 반영



GBM (Gradient Boosting)

AdaBoost와 기본 개념은 동일

가중치 계산 방식이 Gradient Descent 이용 하여 최적의 파라메터 찾기

$$Y = M(x) + \underline{error}$$

$$error = \underline{G(x) + error2}$$

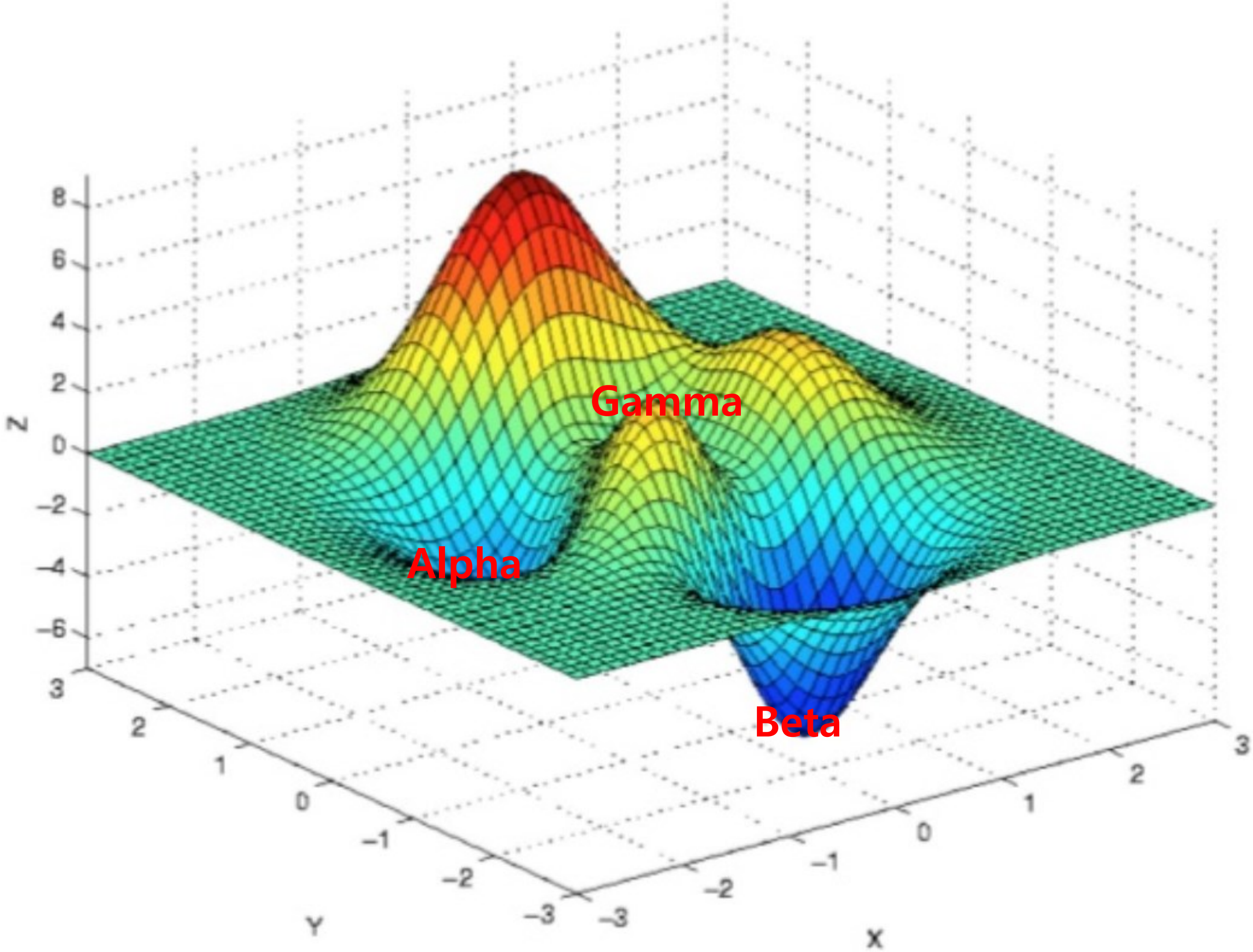
$$error2 = H(x) + error3$$

$$Y = M(x) + \underline{G(x) + H(x) + error3}$$

$$Y = \underline{alpha} * M(x) + \underline{beta} * G(x) + \underline{gamma} * H(x) + error4$$

Gradient Decent

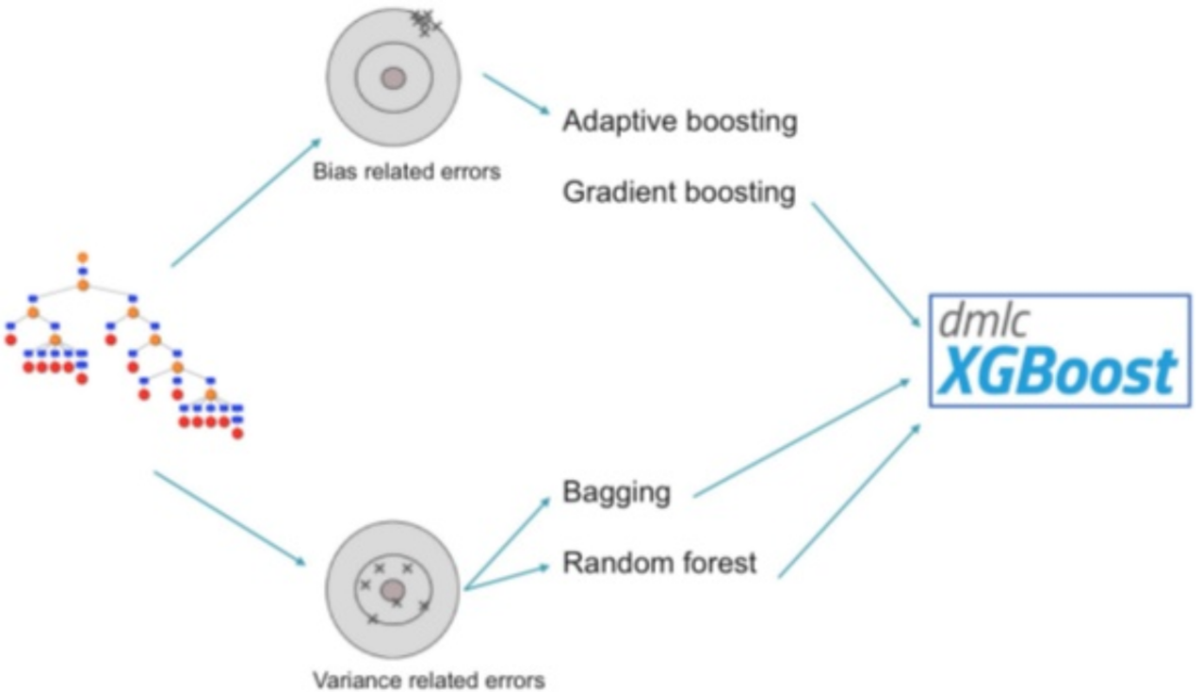
GBM (Gradient Boosting)



Gradient Decent

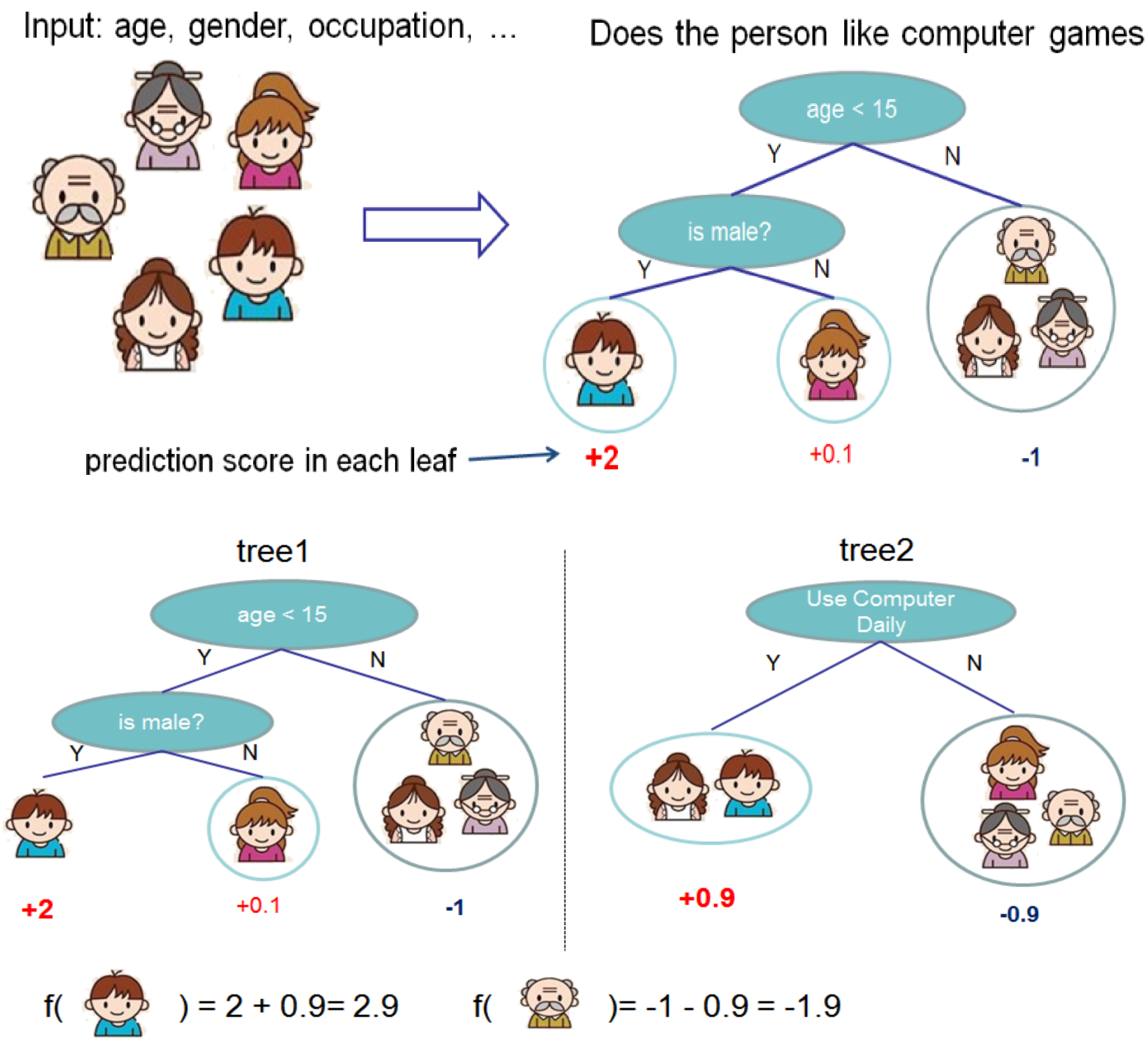
XGBoost (eXtreme Gradient Boosting)

GBM + 분산 / 병렬 처리 지원



XGBoost (eXtreme Gradient Boosting)

CART(Classification And Regression Trees) 집합



XGBoost (eXtreme Gradient Boosting)

Tree를 어떻게 분리 방법

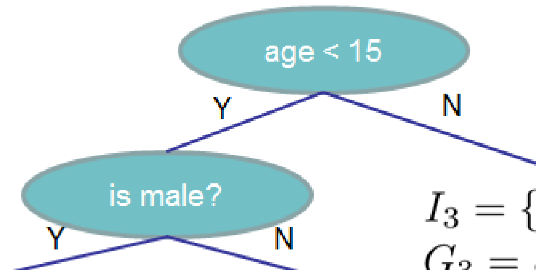
왼쪽 Leaf Score **오른쪽 Leaf Score**

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma$$

Regularization

Gain값이 γ 보다 작으면 분리 하지 않음 **원본 Leaf Score**

Instance index		gradient statistics
1		g1, h1
2		g2, h2
3		g3, h3
4		g4, h4
5		g5, h5



$I_1 = \{1\}$, $G_1 = g_1$, $H_1 = h_1$

$I_2 = \{4\}$, $G_2 = g_4$, $H_4 = h_4$

$I_3 = \{2, 3, 5\}$, $G_3 = g_2 + g_3 + g_5$, $H_3 = h_2 + h_3 + h_5$

$$Obj = -\sum_j \frac{G_j^2}{H_j+\lambda} + 3\gamma$$

The smaller the score is, the better the structure is

XGBoost (eXtreme Gradient Boosting)

Algorithm 1: Exact Greedy Algorithm for Split Finding

Input: I, instance set of current node
Input: d, feature dimension
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i$, $H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ *to* m **do**
  $G_L \leftarrow 0,\ H_L \leftarrow 0$
  for j *in sorted*(I, *by* $\mathbf{x}_{jk}$) **do**
    $G_L \leftarrow G_L + g_j,\ H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L,\ H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split with max score

- For each node, enumerate over all features
  - For each feature, sorted the instances by feature value
  - Use a linear scan to decide the best split along that feature
  - Take the best split solution along all the features

Light GBM

- Decision Tree 알고리즘기반의 GBM 프레임워크 (빠르고, 높은 성능)
- Ranking, classification 등의 문제에 활용

차이점

- Leaf-wise로 tree를 성장(수직 방향) , 다른 알고리즘 (Level-wise)
- 최대 delta loss의 leaf를 성장
- 동일한 leaf를 성장할때, Leaf-wise가 loss를 더 줄일 수 있다.

Light GBM 인기

- 대량의 데이터를 병렬로 빠르게 학습가능 (Low Memory, GPU 활용가능)
- 예측정확도가 더 높음(Leaf-wise tree의 장점 à 과접합에 민감)

속도

- XGBoost 대비 2~10배 (동일한 파라미터 설정시)

사용 빈도

- Light GBM이 설치된 툴이 많이 없음. XGBoost(2014), Light GBM(2016)

활용

- Leaf-wise Tree는 overfitting에 민감하여, 대량의 데이터 학습에 적합
- 적어도 10,000 건 이상

참조자료

https://mlwave.com/kaggle-ensembling-guide/
http://pythonkim.tistory.com/42
https://www.slideshare.net/potaters/decision-forests-and-discriminant-analysis
https://github.com/kaz-Anova/StackNet
https://swalloow.github.io/bagging-boosting
https://www.slideshare.net/freepsw/boosting-bagging-vs-boosting
https://www.youtube.com/watch?v=GM3CDQfQ4sw
http://blog.kaggle.com/2017/06/15/stacking-made-easy-an-introduction-to-stacknet-by-competitions-grandmaster-marios-michailidis-kazanova/
https://www.analyticsvidhya.com/blog/2015/09/complete-guide-boosting-methods/
https://communedeart.com/2017/06/25/xgboost-%EC%82%AC%EC%9A%A9%ED%95%98%EA%B8%B0/
http://xgboost.readthedocs.io/en/latest/model.html

감사합니다.