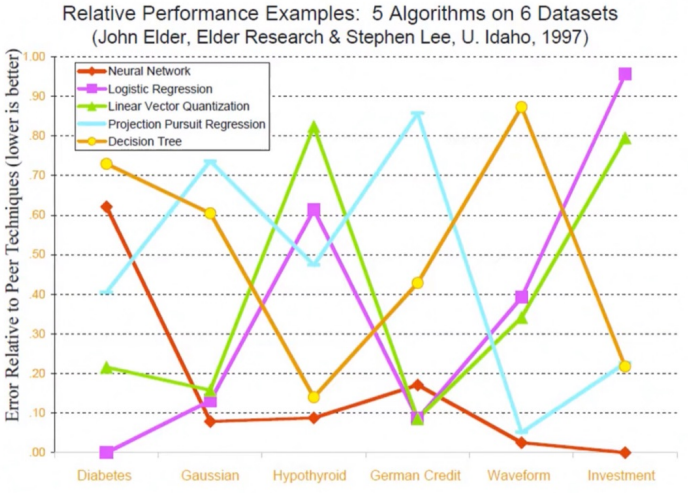
Ensemble Learning.

- Can we have a superior algorithm for all datasets?
  - ✓ Every algorithm scored best or next-to-best on at least two of the six data sets.



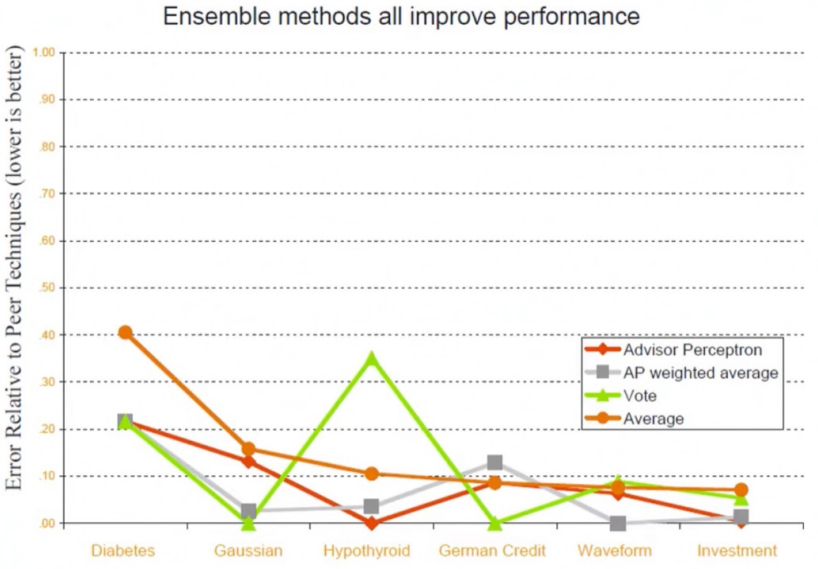
모든 상황에서 우월한 알고리즘? X

⇒ 이것저것 해보는게 훨씬 안전.

(여러 알고리즘을 공부하는 이유)

However, if they are properly combined ...

$\geq$ best of individual algorithms.



No Free Lunch Theorem.

179 algorithms & 121 data sets.

Journal of Machine Learning Research 15 (2014) 3133-3181 Submitted 11/13; Revised 4/14; Published 10/14

Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?

Manuel Fernández-Delgado MANUEL.FERNANDEZ.DELGADO@USC.ES
Eva Cernadas EVA.CERNADAS@USC.ES
Senén Barro SENEN.BARRO@USC.ES
CITIUS: Centro de Investigación en Tecnoloxías da Información da USC
University of Santiago de Compostela
Campus Vida, 15872, Santiago de Compostela, Spain

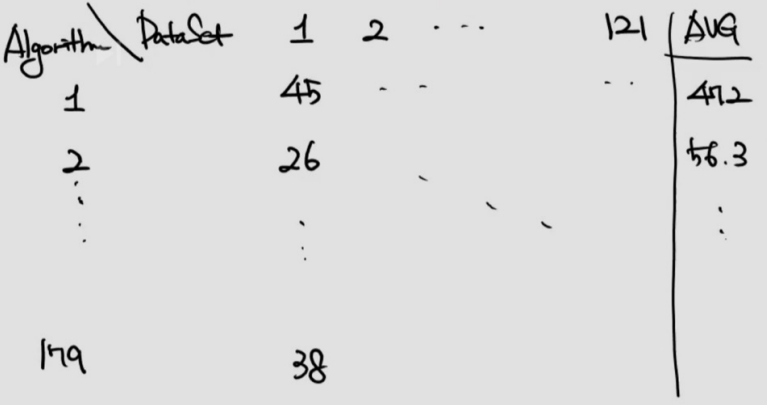
Dinani Amorim DINANIAMORIM@GMAIL.COM
Departamento de Tecnologia e Ciências Sociais- DTCS
Universidade do Estado da Bahia
Av. Edgard Chastinet S/N - São Geraldo - Juazeiro-BA, CEP: 48.305-680, Brasil

Empirical Evidence

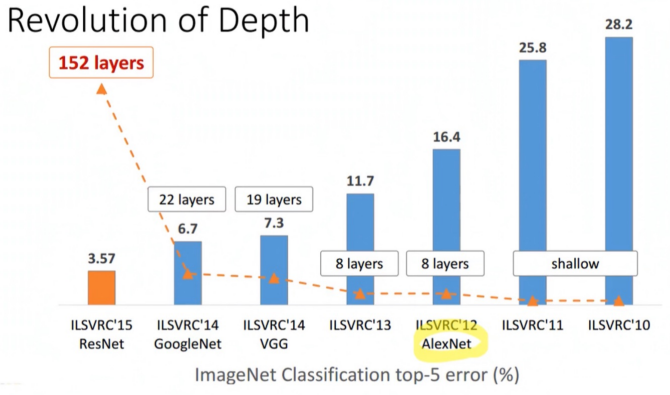
- Empirical study 3: 179 algorithms on 121 datasets

Rank	Acc.	κ	Classifier	Rank	Acc.	κ	Classifier
32.9	82.0	63.5	parRF_t (RF)	67.3	77.7	55.6	pda_t (DA)
33.1	**82.3**	**63.6**	rf_t (RF)	67.6	78.7	55.2	elm_m (NNET)
36.8	81.8	62.2	svm_C (SVM)	67.6	77.8	54.2	SimpleLogistic_w (LMR)
38.0	81.2	60.1	svmPoly_t (SVM)	69.2	78.3	57.4	MAB_J48_w (BST)
39.4	81.9	62.5	rforest_R (RF)	69.8	78.8	56.7	BG_REPTree_w (BAG)
39.6	82.0	62.0	elm_kernel_m (NNET)	69.8	78.1	55.4	SMO_w (SVM)
40.3	81.4	61.1	svmRadialCost_t (SVM)	70.6	78.3	58.0	MLP_w (NNET)
42.5	81.0	60.0	svmRadial_t (SVM)	71.0	78.8	58.23	BG_RandomTree_w (BAG)
42.9	80.6	61.0	C5.0_t (BST)	71.0	77.1	55.1	mlm_R (GLM)
44.1	79.4	60.5	avNNet_t (NNET)	71.0	77.8	56.2	BG_J48_w (BAG)
45.5	79.5	61.0	nnet_t (NNET)	72.0	75.7	52.6	rbf_t (NNET)
47.0	78.7	59.4	pcaNNet_t (NNET)	72.1	77.1	54.8	fda_R (DA)
47.1	80.8	53.0	BG_LibSVM_w (BAG)	72.4	77.0	54.7	lda_R (DA)
47.3	80.3	62.0	mlp_t (NNET)	72.4	79.1	55.6	svmlight_C (NNET)
47.6	80.6	60.0	RotationForest_w (RF)	72.6	78.4	57.9	AdaBoostM1_J48_w (BST)
50.1	80.9	61.6	RRF_t (RF)	72.7	78.4	56.2	BG_IBk_w (BAG)
51.6	80.7	61.4	RRFglobal_t (RF)	72.9	77.1	54.6	ldaBag_R (BAG)
52.5	80.6	58.0	MAB_LibSVM_w (BST)	73.2	78.3	56.2	BG_LWL_w (BAG)
52.6	79.9	56.9	LibSVM_w (SVM)	73.7	77.9	56.0	MAB_REPTree_w (BST)
57.6	79.1	59.3	adaboost_R (BST)	74.0	77.4	52.6	RandomSubSpace_w (DT)
58.5	79.7	57.2	pnn_m (NNET)	74.4	76.9	54.2	lda2_t (DA)
58.9	78.5	54.7	cforest_t (RF)	74.6	74.1	51.8	svmBag_R (BAG)
59.9	79.7	42.6	dkp_C (NNET)	74.6	77.5	55.2	LibLINEAR_w (SVM)
60.4	80.1	55.8	gausspr RadialLR (OM)	75.9	77.2	55.6	rbfDDA_t (NNET)
60.5	80.0	57.4	RandomForest_w (RF)	76.5	76.9	53.8	sda_t (DA)
62.1	78.7	56.0	svmLinear_t (SVM)	76.6	78.1	56.5	END_w (OEN)
62.5	78.4	57.5	fda_t (DA)	76.6	77.3	54.8	LogitBoost_w (BST)
62.6	78.6	56.0	knn_t (NN)	76.6	78.2	57.3	MAB_RandomTree_w (BST)
62.8	78.5	58.1	mlp_C (NNET)	77.1	78.4	54.0	BG_RandomForest_w (BAG)
63.0	79.9	59.4	RandomCommittee_w (OEN)	78.5	76.5	53.7	Logistic_w (LMR)
63.4	78.7	58.4	Decorate_w (OEN)	78.7	76.6	50.5	ctreeBag_R (BAG)
63.6	76.9	56.0	mlpWeightDecay_t (NNET)	79.0	76.8	53.5	BG_Logistic_w (BAG)
63.8	78.7	56.7	rda_R (DA)	79.1	77.4	53.0	lvq_t (NNET)
64.0	79.0	58.6	MAB_MLP_w (BST)	79.1	74.4	50.7	pls_t (PLSR)
64.1	79.9	56.9	MAB_RandomForest_w (BST)	79.8	76.9	54.7	hdda_R (DA)
65.0	79.0	56.8	knn_R (NN)	80.6	75.9	53.3	MCC_w (OEN)
65.2	77.9	56.2	multinom_t (LMR)	80.9	76.9	54.5	mda_R (DA)
65.5	77.4	56.6	gcvEarth_t (MARS)	81.4	76.7	55.2	C5.0Rules_t (RL)
65.5	77.8	55.7	glmnet_R (GLM)	81.6	78.3	55.8	lssvmRadial_t (SVM)
65.6	78.6	58.4	MAB_PART_w (BST)	81.7	75.6	50.9	JRip_t (RL)
66.0	78.5	56.5	CVR_w (OM)	82.0	76.1	53.3	MAB_Logistic_w (BST)
66.4	79.2	58.9	treebag_t (BAG)	84.2	75.8	53.9	C5.0Tree_t (DT)
66.6	78.2	56.8	BG_PART_w (BAG)	84.6	75.7	50.8	BG_DecisionTable_w (BAG)
66.7	75.5	55.2	mda_t (DA)	84.9	76.5	53.4	NBTree_w (DT)

Rank	Acc.	κ	Classifier
32.9	82.0	63.5	parRF_t (RF)
33.1	**82.3**	**63.6**	rf_t (RF)
36.8	81.8	62.2	svm_C (SVM)
38.0	81.2	60.1	svmPoly_t (SVM)
39.4	81.9	62.5	rforest_R (RF)



- Large Scale Visual Recognition Challenge (~ ILSVRC2015)



✓2016

Object detection (DET)[ilsvrc]

Task 1a: Object detection with provided training data

Ordered by number of categories won

Team name	Entry description	Number of object categories won	mean AP
CUImage	Ensemble of 6 models using provided data	109	0.662751
Hikvision	Ensemble A of 3 RPN and 6 FRCN models, mAP is 67 on val2	30	0.652704
Hikvision	Ensemble B of 3 RPN and 5 FRCN models, mean AP is 66.9, median AP is 69.3 on val2	18	0.652003

Object localization (LOC)[ilsvrc]

Task 2a: Classification+localization with provided training data

Ordered by localization error

Team name	Entry description	Localization error	Classification error
Trimps-Soushen	Ensemble 3	0.077087	0.02991
Trimps-Soushen	Ensemble 4	0.077429	0.02991
Trimps-Soushen	Ensemble 2	0.077668	0.02991
Trimps-Soushen	Ensemble 1	0.079068	0.03144

양상블의 시대

✓2017

Object detection (DET)[ilsvrc]

Task 1a: Object detection with provided training data

Ordered by number of categories won

Team name	Entry description	Number of object categories won	mean AP
BDAT	submission4	85	0.731592
BDAT	submission3	65	0.732227
BDAT	submission2	30	0.723712
DeepView(ETRI)	Ensemble_A	10	0.593084
NUS-Qihoo_DPNs (DET)	Ensemble of DPN models	9	0.656932
KAISTNIA_ETRI	Ensemble Model5	1	0.61022
KAISTNIA_ETRI	Ensemble Model4	0	0.609402
KAISTNIA_ETRI	Ensemble Model2	0	0.608299
KAISTNIA_ETRI	Ensemble Model1	0	0.608278
KAISTNIA_ETRI	Ensemble Model3	0	0.60631

Object localization (LOC)[ilsvrc]

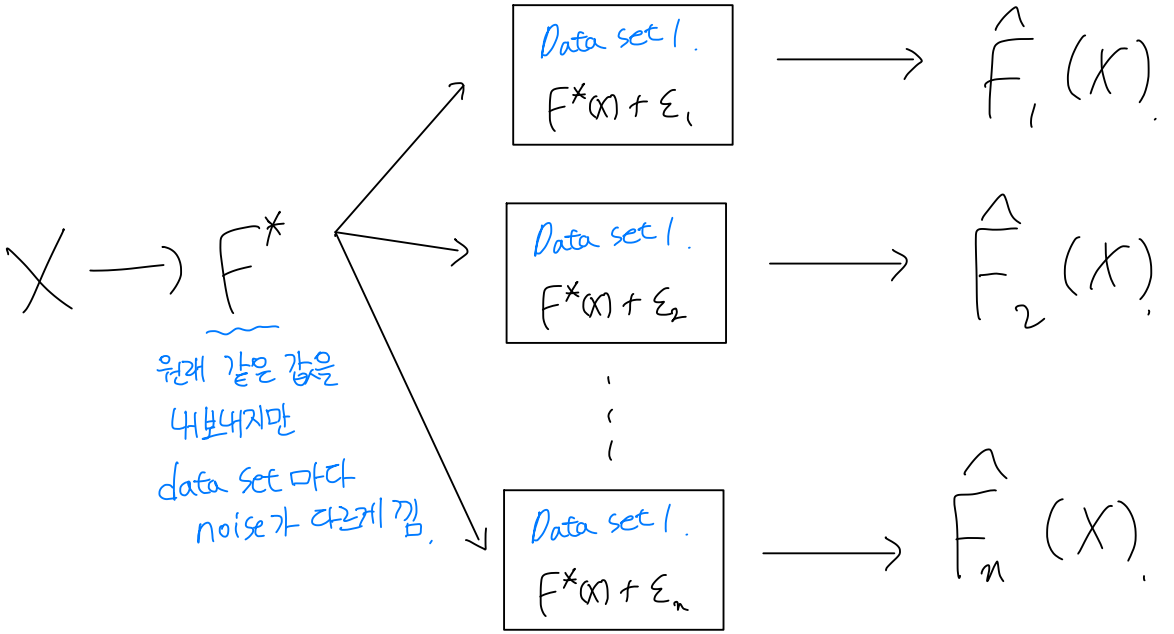
Task 2a: Classification+localization with provided training data

Ordered by localization error

Team name	Entry description	Localization error	Classification error
NUS-Qihoo_DPNs (CLS-LOC)	[E3] LOC: Dual Path Networks + Basic Ensemble	0.062263	0.03413
Trimps-Soushen	Result-3	0.064991	0.02481
Trimps-Soushen	Result-2	0.06525	0.02481
Trimps-Soushen	Result-4	0.065261	0.02481
Trimps-Soushen	Result-5	0.065302	0.02481
Trimps-Soushen	Result-1	0.067698	0.02481

Bias - Variance Decomposition.

$$y = F^*(x) + \varepsilon \quad , \quad \varepsilon \sim N(0, \sigma^2)$$



$$\bar{F} = E[\hat{F}_D(X)]$$

$$Err(x_0) = E\left[y - \hat{f}(x) \mid x = x_0 \right]$$

(MSE)

$$= E\left[f^*(x_0) + \varepsilon - \hat{f}(x_0) \right]^2 \qquad (y = f^*(x) + \varepsilon)$$

$$= E\left[f^*(x_0) - \hat{f}(x_0) \right]^2 + \sigma^2$$

$$= E\left[\underbrace{f^*(x_0) - \bar{f}(x_0)}_{A} + \underbrace{\bar{f}(x_0) - \hat{f}(x_0)}_{B} \right]^2 + \sigma^2$$

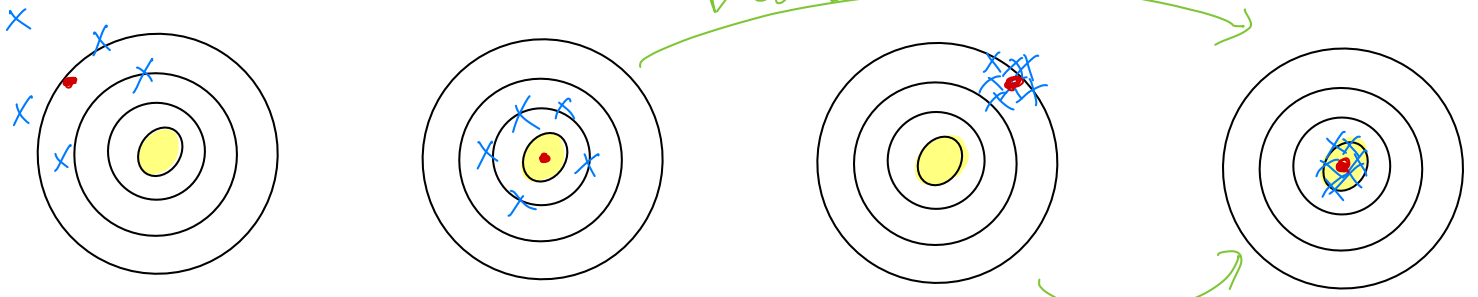
$$\left(2E\left[\left(f^*(x_0) - \bar{f}(x_0) \right)\left(\bar{f}(x_0) - \hat{f}(x_0) \right) \right] = 0 \right)$$

$$= E\left[F^*(x_0) - \bar{F}(x_0) + \bar{F}(x_0) - \hat{F}(x_0) \right]^2 + \sigma^2$$

$$= E\left[F^*(x_0) - \bar{F}(x_0) \right]^2 + E\left[\bar{F}(x_0) - \hat{F}(x_0) \right]^2 + \sigma^2$$

$$= \left[F^*(x_0) - \bar{F}(x_0) \right]^2 + E\left[\bar{F}(x_0) - \hat{F}(x_0) \right]^2 + \sigma^2$$

$$= Bias^2(\hat{F}(x_0)) + Var(\hat{F}(x_0)) + \sigma^2$$



Bias	High	Low	High	Low
Variance.	High	High	Low	Low.

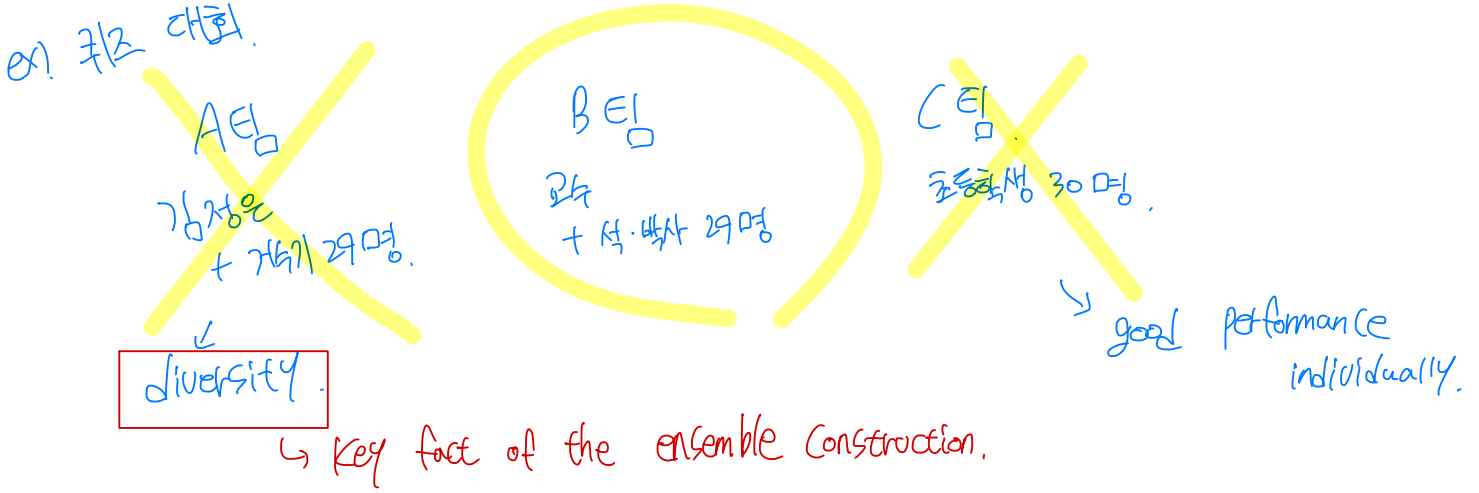
complexity ↑
model.

Purpose of Ensemble.

Reduce the error through constructing multiple learners.

Var ↓ : Bagging., Random Forests. (≠ Bagging)

Bias ↓ : Boosting



Average Error vs Ensemble Error.

$$E_{Avg} = \frac{1}{M}\sum_{m=1}^{M} E\left[\epsilon_m(x)^2\right]$$

앙상블의 출력: 개별 모델들의 평균.

$$E_{Ensemble} = E\left[\left\{\frac{1}{M}\sum_{m=1}^{M} y_m(x) - f(x)\right\}^2\right]$$

$$= E\left[\left\{\frac{1}{M}\cdot\left(\sum_{m=1}^{M} y_m(x) - M\cdot f(x)\right)\right\}^2\right]$$

$$= E\left[\left\{\frac{1}{M}\cdot\sum_{m=1}^{M}\left(y_m(x) - f(x)\right)\right\}^2\right]$$

$$= E\left[\left\{\frac{1}{M}\cdot\sum_{m=1}^{M}\epsilon_m(x)\right\}^2\right].$$

$$= E\left[\left\{\frac{1}{M}\cdot\sum_{m=1}^{M}\varepsilon_m(x)\right\}^2\right].$$

Assume that zero mean & uncorrelated.

$$E[\varepsilon_m(x)]=0 \qquad E[\varepsilon_m(x)\cdot\varepsilon_\ell(x)]=0 \quad (m\neq\ell).$$

$$\Rightarrow E_{Ensemble} = \frac{1}{M}\cdot E_{AVG}.$$

(But in reality errors are correlated.).

In reality, by Cauchy inequality

$$(a^2+b^2)(x^2+y^2) \geq (ax+by)^2.$$

$$\left\{1\cdot\varepsilon_1(x) + 1\cdot\varepsilon_2(x) + \cdots + 1\cdot\varepsilon_M(x)\right\}^2 \leq \overbrace{(1+1+\cdots+1)}^{M}\left(\varepsilon_1(x)^2+\varepsilon_2(x)^2+\cdots+\varepsilon_M(x)^2\right)$$

$$\hookrightarrow \left(\sum_{m=1}^{M}\varepsilon_m(x)\right)^2 \qquad = M\cdot\sum_{m=1}^{M}\varepsilon_m(x)^2$$

$$\Rightarrow \left\{\frac{1}{M}\sum_{m=1}^{M}\varepsilon_m(x)\right\}^2 \leq \frac{1}{M}\cdot\sum_{m=1}^{M}\varepsilon_m(x)^2.$$

$$\Rightarrow E_{\text{Ensemble}} \leq E_{\text{Avg}}.$$

← 앙상블의 핵심은 모델들의 "diversity"

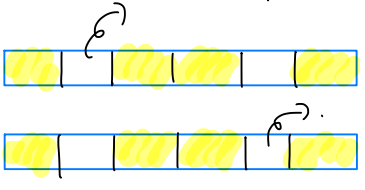
Bagging.

diversity 주는 방법.

- Data. → Bagging.
- Model. → Random Forests.

i) k-fold.

- Sampling with replacement.



- F_i
- F_j.

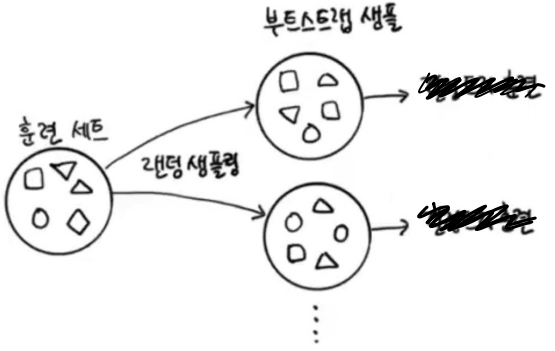
F_i, F_j 는 (k-2) folds 공유. ⇒ 상관.

ii) Bagging (Bootstrap Aggregating)

: Choosing N uniformly random with replacement.

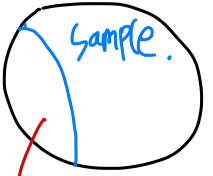
Bootstrap Sampling

↳ 복원 추출.



이론적으로 한 번도 선택되지 않을 확률.

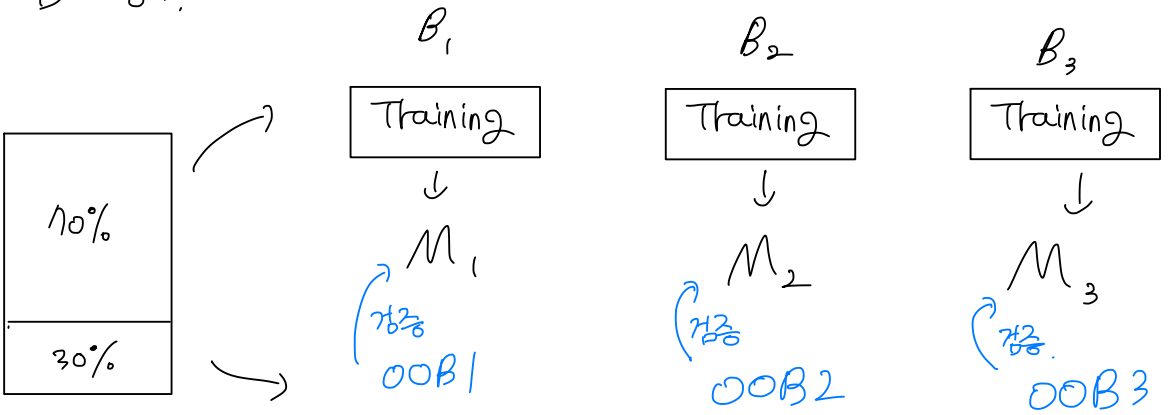
$$(1-\frac{1}{N})^N \Rightarrow \lim_{N\to\infty}(1-\frac{1}{N})^N = e^{-1} = 0.368$$



↳ OOB. (out of bag)

검증세트로 활용 가능.

OOB 평가

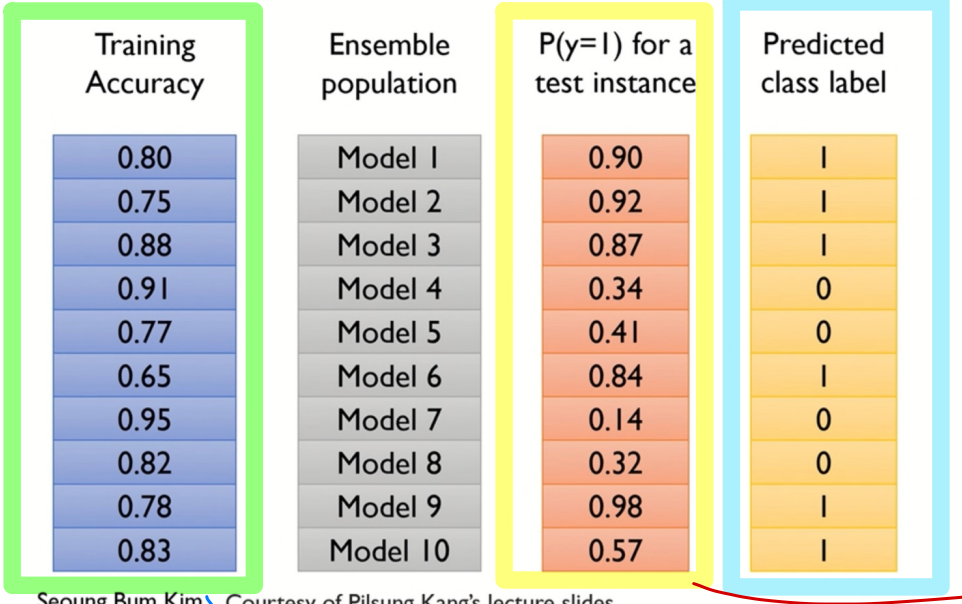


개별적인 객체들이

어떤 경우에는 학습데이터

어떤 경우에는 검증 데이터.

Result Aggregating.



Training Accuracy	Ensemble population	P(y=1) for a test instance	Predicted class label
0.80	Model 1	0.90	1
0.75	Model 2	0.92	1
0.88	Model 3	0.87	1
0.91	Model 4	0.34	0
0.77	Model 5	0.41	0
0.65	Model 6	0.84	1
0.95	Model 7	0.14	0
0.82	Model 8	0.32	0
0.78	Model 9	0.98	1
0.83	Model 10	0.57	1

Seoung Bum Kim Courtesy of Pilsung Kang's lecture slides

라벨의 비율.

1 - 6개 → 1 예측.

0 - 4개

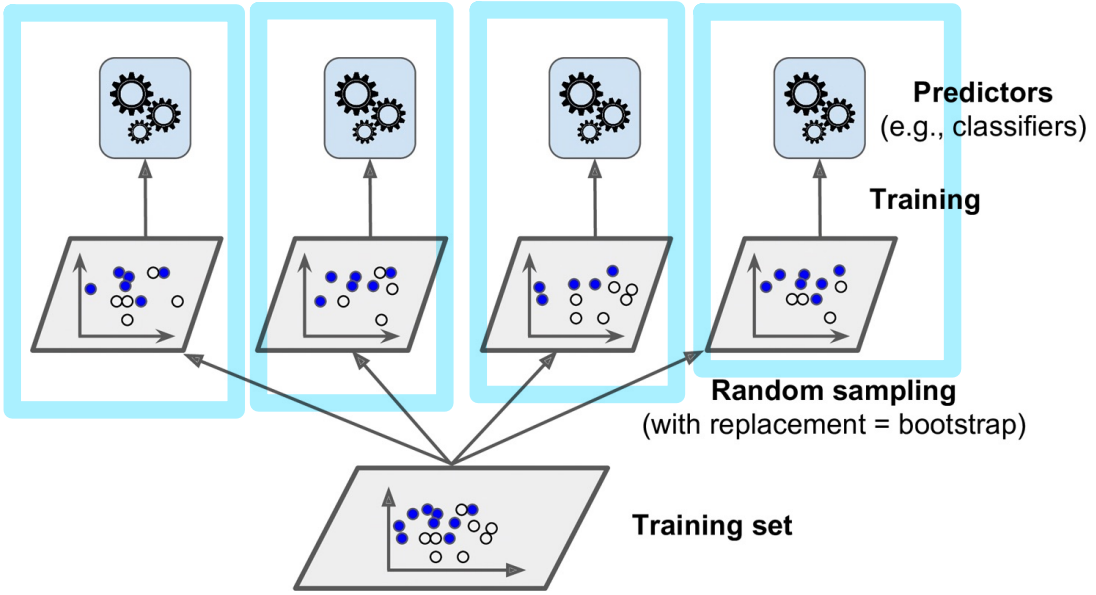
정확도 가중치 적용

$1 \times \square + 1 \times \square + \cdots$

10.

확률의 평균 값.

동시에 훈련 가능!

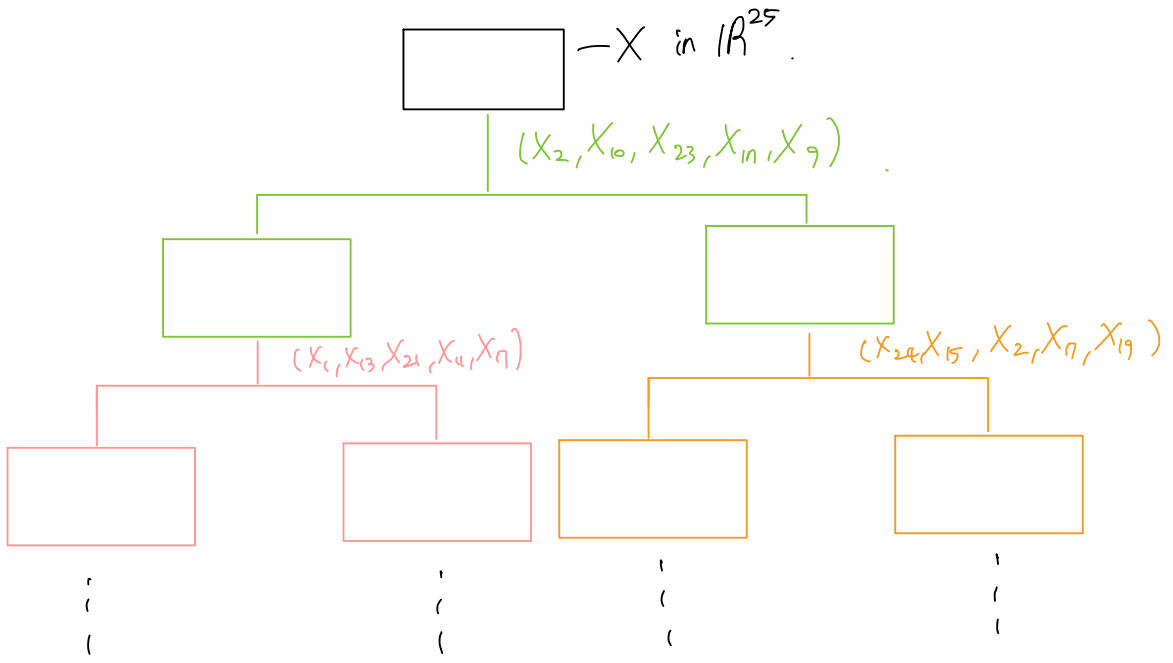

Predictors
(e.g., classifiers)
Training
Random sampling
(with replacement = bootstrap)
Training set

Random Forests.

Bagging + Decision Tree (with random subspace).

Data를 이용한 diversity.

Model을 이용한 diversity.



(diversity ↑ performance ↓.

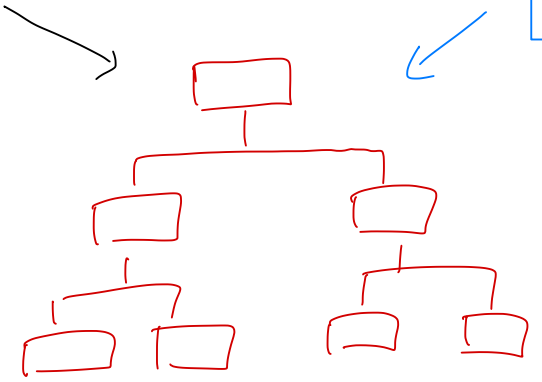
Feature Importance.

Original OOB Data.

	X_i	
	1	
	2	
	3	
	4	
	5	

(after permutation X_i) OOB Data.

	X_i	
	2	
	5	
	3	
	1	
	4	



OOB Error of the original Data e_i.

OOB Error of the permutated Data P_i.

변수의 중요도가 높다면?

- Permutation 전·후 OOB Error 차이가 커야 함.
- 그 차이의 편차가 적어야 함.

Let $d_i^m = p_i^m - e_i^m$ (m번째 tree, X_i에 대한 OOB error 차이)

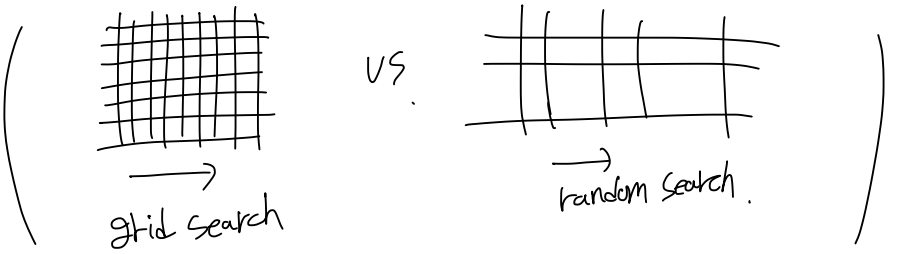
$\bar{d}_i = \frac{1}{M}\sum_{m=1}^{M} d_i^m$, $s_i^2 = \frac{1}{M-1}\sum_{m=1}^{M}(d_i^m - \bar{d}_i)^2$

$\Rightarrow$ Feature importance of X_i $\quad V_i = \frac{\bar{d}_i}{s_i}$

Extra Trees.

Model diversity ↑

random subspace
+
radom search.

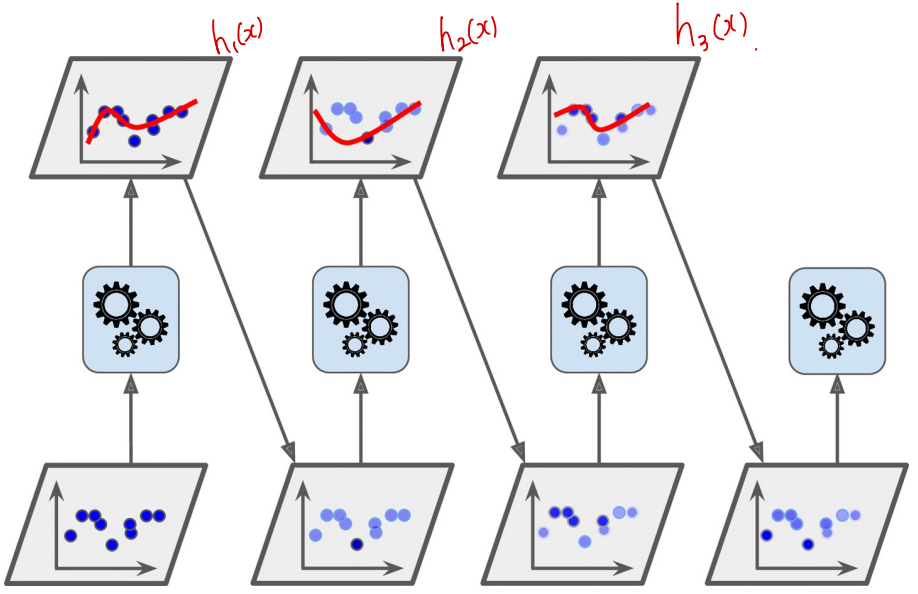


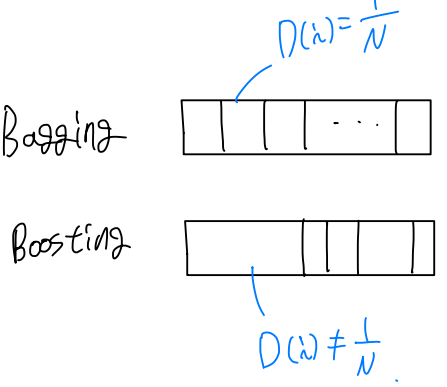
정확도 ↑

속도 ↑
다양성 ↑

+ bootstrap sampling X.

Boosting.





$$H(x) = \text{sign}\left(\alpha_1 h_1(x) + \alpha_2 \cdot h_2(x) + \cdots + \alpha_k \cdot h_k(x)\right).$$

Ada Boosting.

Training Data $S = \{ (x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N) \}$ where $y_n = (-1, 1)$.

Define uniform distribution $D_1(i)$ over elements of S.

$D_1(i)$: 첫번째 Data set에 example i가 선택될 확률.

i) Train h_t using distribution D_t.

ii) Calculate $\varepsilon_t = P_{D_t}(h_t(x) \neq y)$ 오분류율. ($\varepsilon_t > 0.5$: stop)

set $\alpha_t = \frac{1}{2} \cdot \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$.

α_t
- $\varepsilon_t = 0.5 \rightarrow \frac{1}{2}\ln\left(\frac{0.5}{0.5}\right) = 0$.
- $\varepsilon_t = 0 \rightarrow \frac{1}{2}\ln\left(\frac{1}{0}\right) = \nearrow$

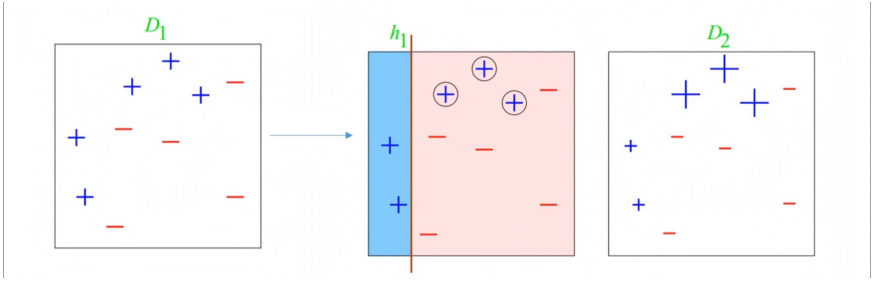
iii) Update $D_{t+1}(i) = \dfrac{D_t(i) \cdot \exp(-\alpha_t \cdot y_t \cdot h_t(x_i))}{Z_t}$

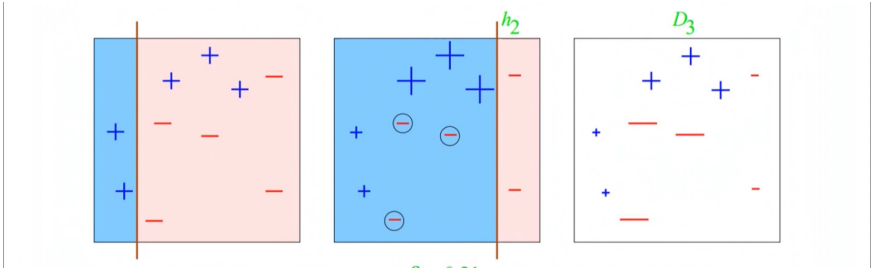
$\dfrac{1}{Z_t} \times D_t(i) \times \exp(-\alpha_t \cdot y_t \cdot h_t(x_i))$ ($y_t \cdot h_t(x_i)$: 1 or -1)

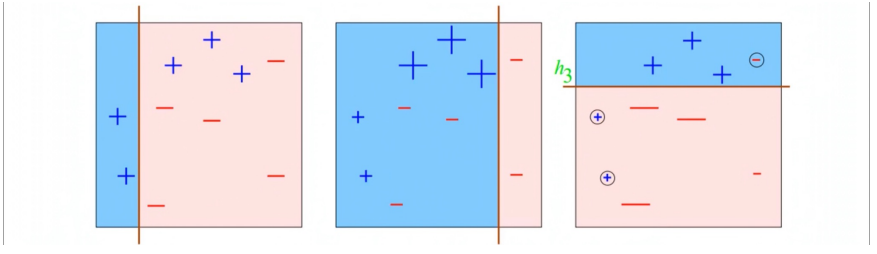
- $\frac{1}{Z_t}$: normalize factor
- $D_t(i)$: t 시점 선택확률의 기준.
- $-\alpha_t$: h_t가 정확하면 sample rate의 변동성 ↑

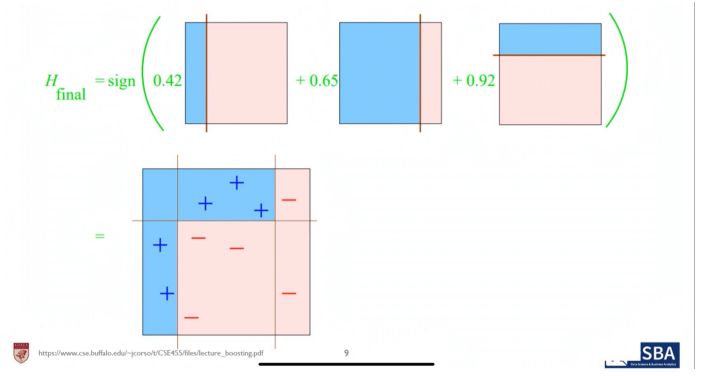
$\Rightarrow$

- h_t가 i의 정답 O → $D_{t+1}(i)$ ↓ ($e^{-\alpha_t}$)
- h_t가 i의 정답 X → $D_{t+1}(i)$ ↑ (e^{α_t})

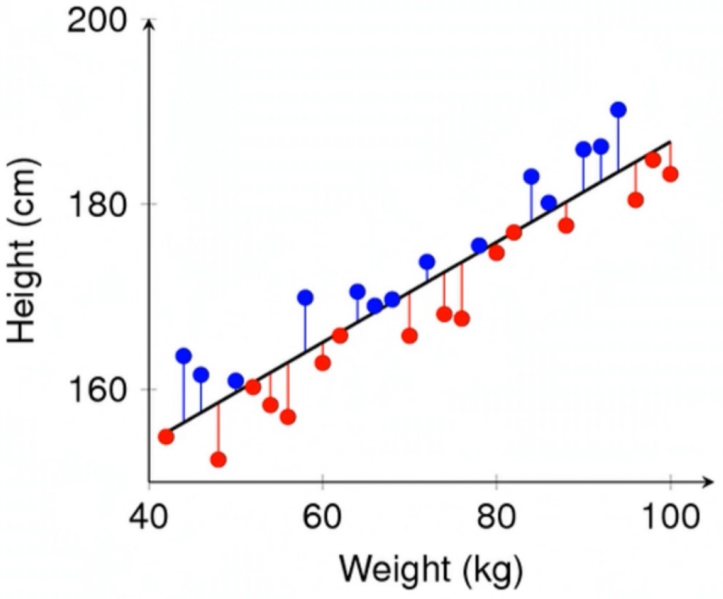

D_1
h_1
D_2


h_2
D_3


h_3


H_{final} = sign
0.42
+ 0.65
+ 0.92
=
https://www.cse.buffalo.edu/~jcorso/t/CSE455/files/lecture_boosting.pdf
9
SBA

Gradient Boosting.



$y = f_1(x)$.

$y - f_1(x) = f_2(x)$.

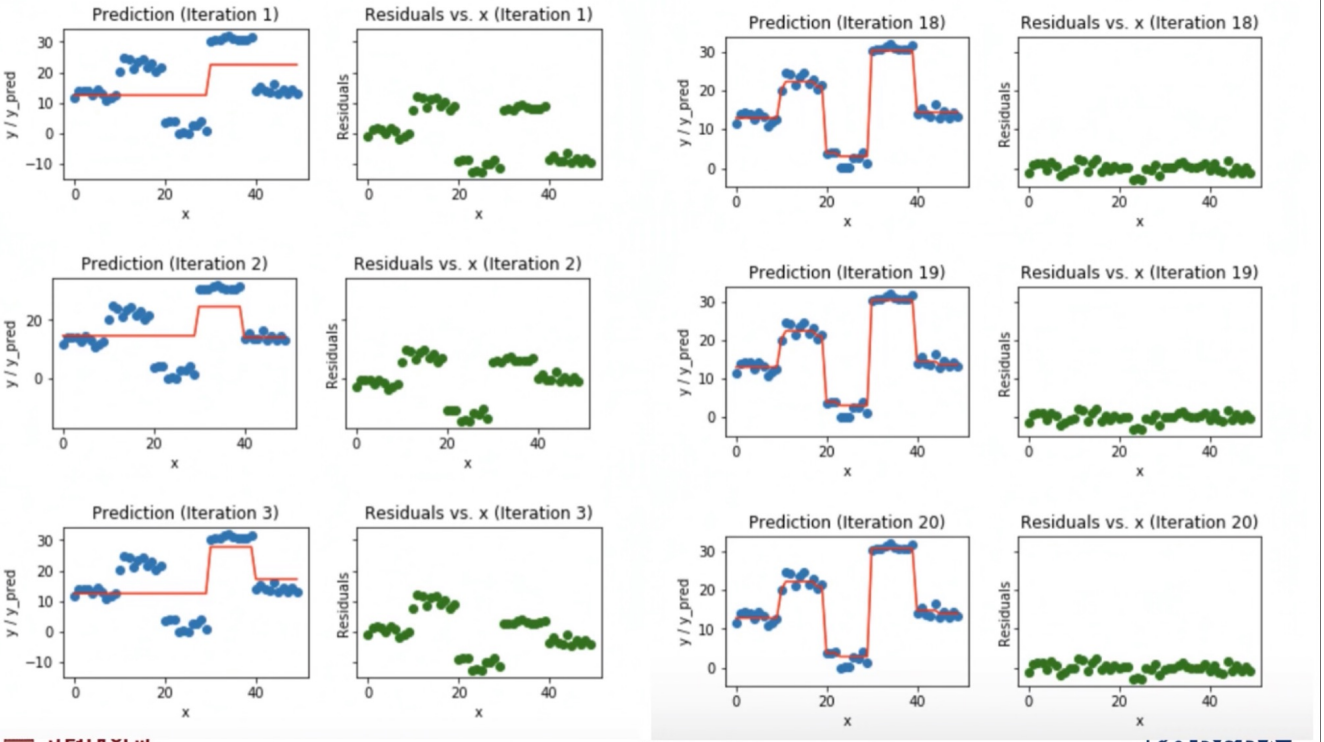
residual

$\Rightarrow$ $y = f_1(x) + f_2(x)$.

(Ada Boosting은 다르게 Data sampling X)

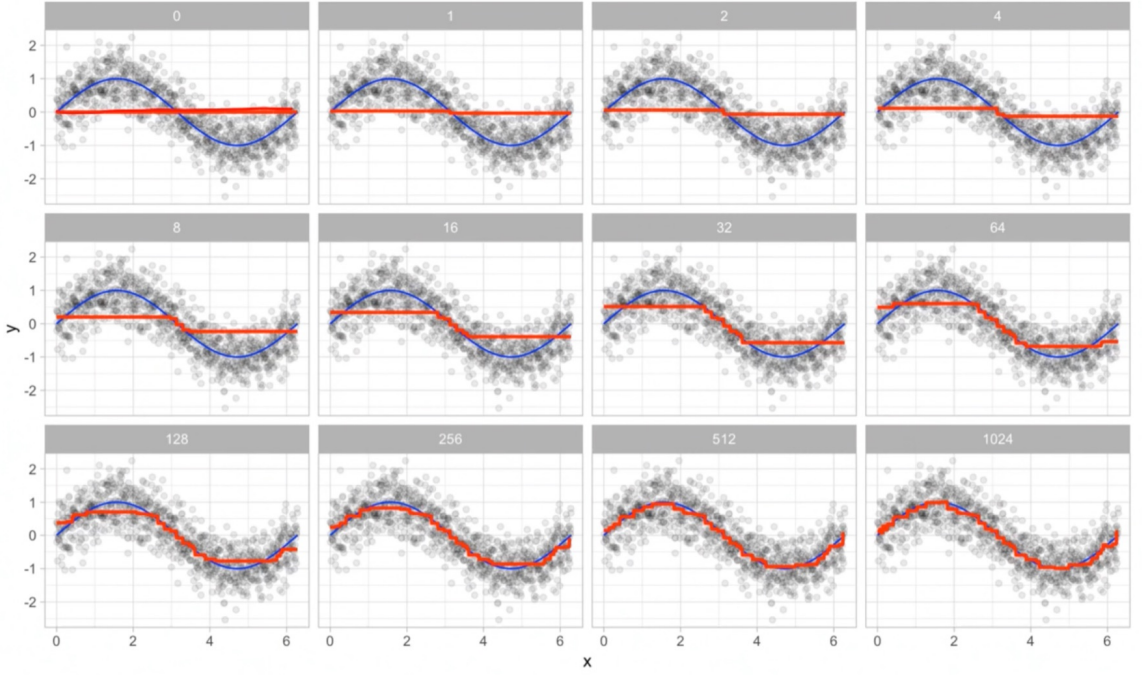
Gradient Boosting Machine: GBM

- GBM Regression Example 1



Gradient Boosting Machine: GBM

- GBM Regression Example 3



https://docs.paperspace.com/machine-learning/wiki/gradient-boosting

#over fitting problem in GBM.

$f_1(x) = y + \varepsilon$

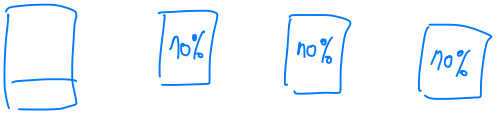
$f_2(x) = (y - f_1(x)) + \varepsilon$

$f_3(x) = (y - f_1(x) - f_2(x)) + \varepsilon$

⋮

↓ ε 비중 ↗ (ε 까지 학습).

=> Regulation.

- Subsampling - 

- shrinkage - $y = f_1(x) + 0.9\, f_2(x) + 0.9^2 f_3(x) + \cdots$
- Early stopping. - 

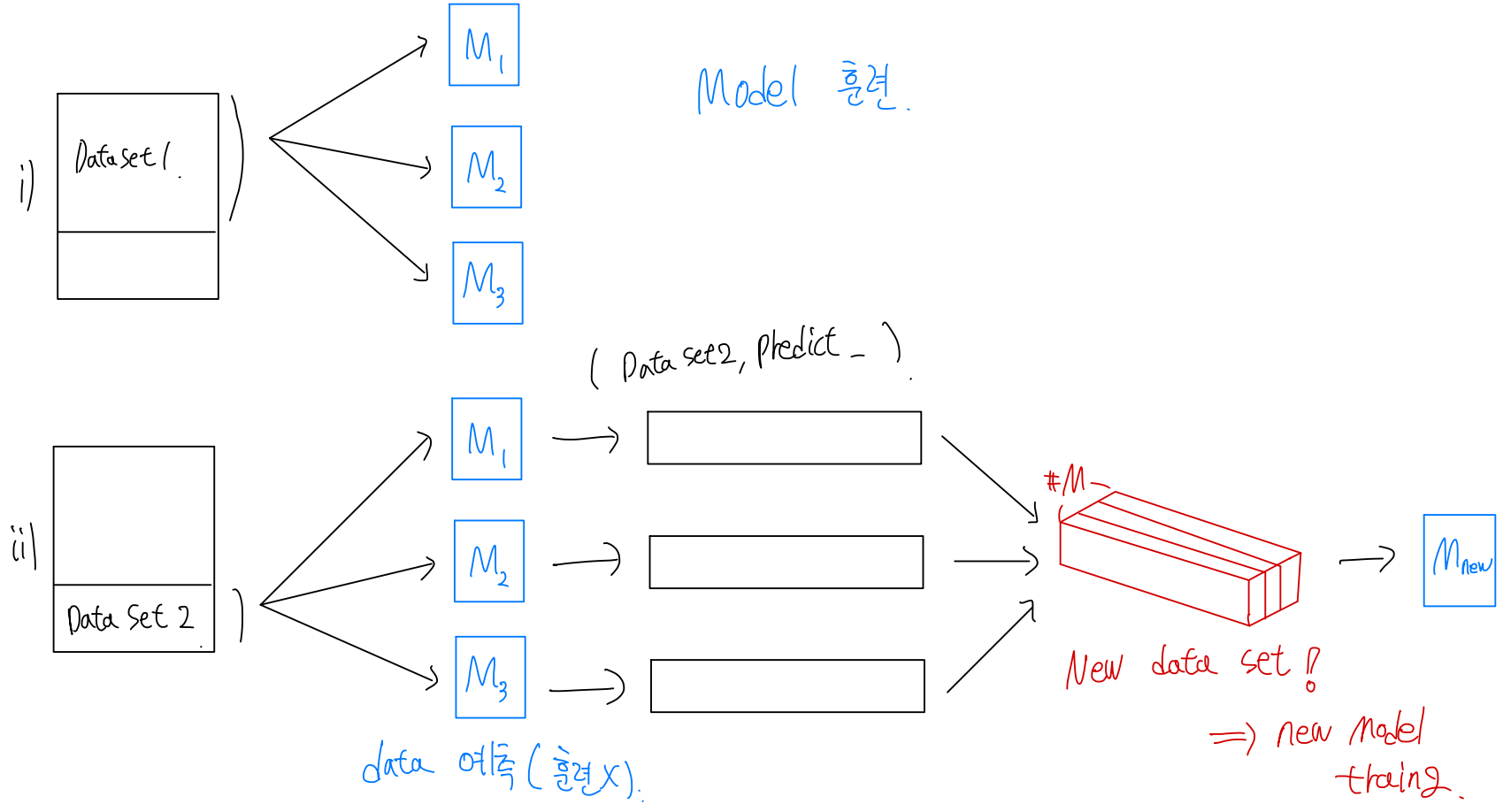
- Variable Importance in Tree-based Gradient Boosting
  - ✓ $Influence_j(T)$: importance of the variable j in a single tree T.
  - ✓ Assume that there are L terminal nodes → $L-1$ splits.

$$Influence_j(T) = \sum_{i=1}^{L-1} (IG_i \times \mathbf{1}(S_i = j))$$

  - ✓ Variable importance of Gradient boosting

$$Influence_j = \frac{1}{M} \sum_{k=1}^{M} Influence_j(T_k)$$

Stacking (앙상블의 앙상블)



Q & A .

유튜브 강의 & 사진 참고

Pilsung Kang
School of Industrical Management Engineering
Korea University