**Gradient boosting** - baf

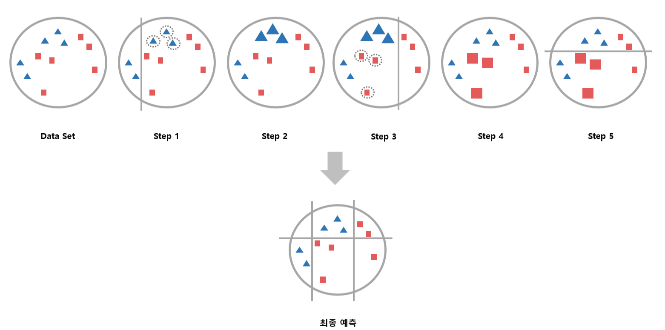
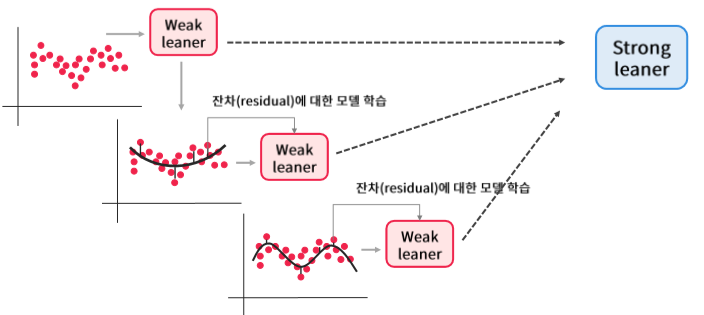
**<앙상블>**

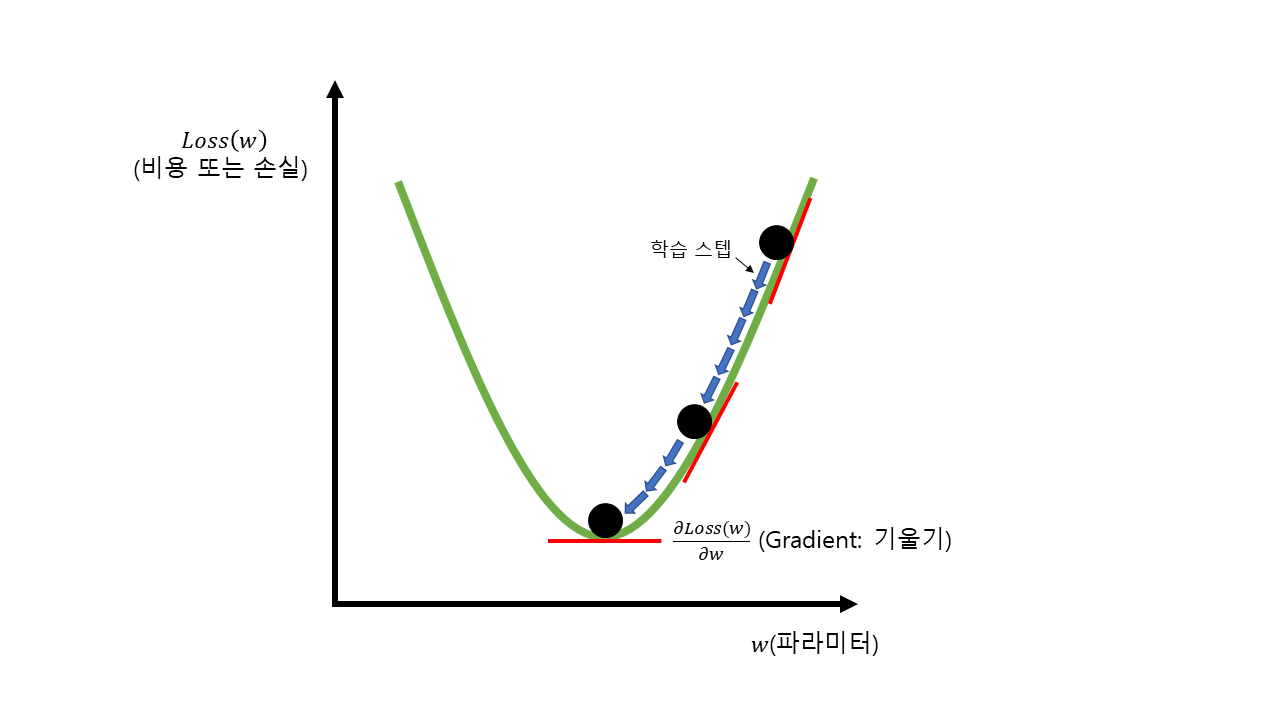
* **voting**
* **bagging**
* **boosting**
* **stacking**

**<Wikipedia-boosting>**

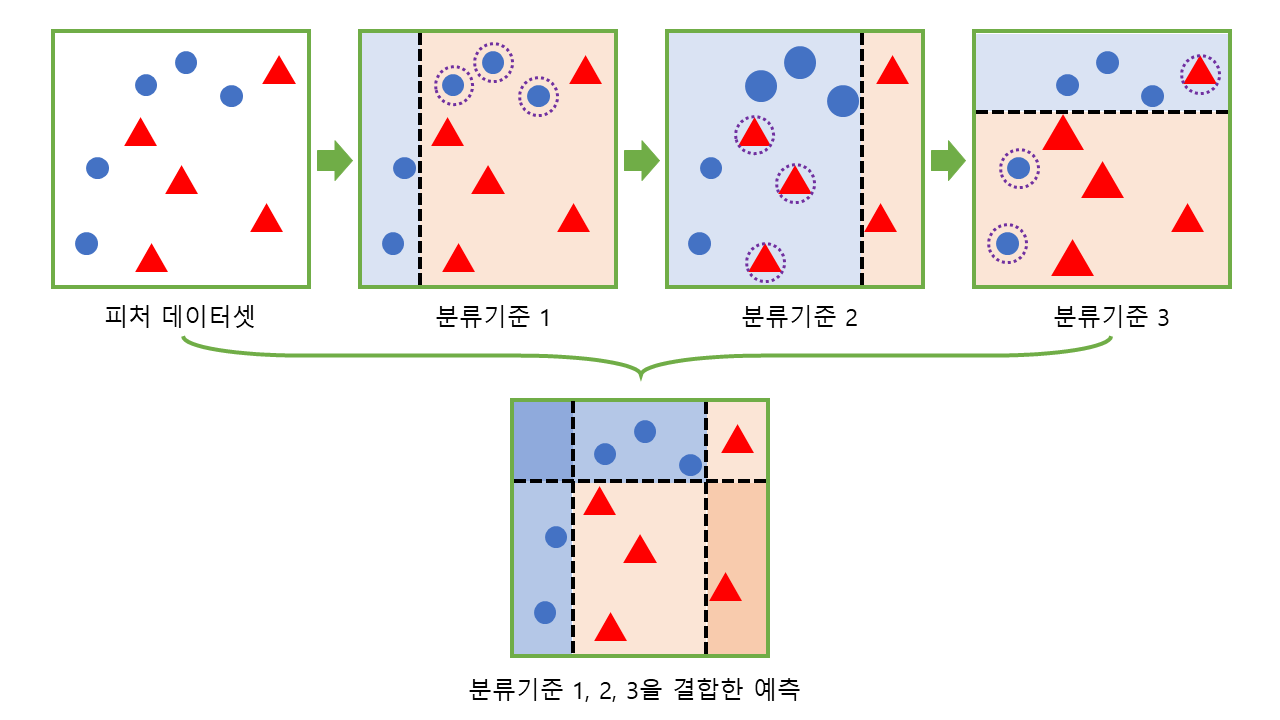
machine learning technique based on boosting in a functional space, where the target is pseudo-residuals rather than the typical residuals used in traditional boosting.

**<boosting & gradient boosting *figures*>**





**<AdaBoost** ***figures*>**

****

AdaBoost는 오류 데이터에 가중치를 부여하면서 부스팅 수행

**<boosting>**

부스팅 알고리즘은 여러 개의 약한 학습기를 순차적으로 학습

예측하며 잘못 예측한 데이터에 가중치를 부여하여 오류를 개선해 나가면서 학습

**<gradient boosting>**

부스팅 알고리즘의 대표적인 모델

AdaBoost와는 다르게 Gradient Descent(경사 하강법)을 이용하여 가중치 업데이트

경사 하강법은 기울기를 줄여나감으로써 오류를 최소화하는 방법

실제값과 모델의 예측값 차이에 따른 오류값은 잔차(Residual)

잔차의 제곱합인 RSS를 비용 함수 또는 손실 함수(loss function)라고 부름

결과적으로 비용함수 값을 지속적으로 감소시켜 최종적으로 더 이상 감소하지 않는 최소의 오류값을 구하는 방향

대략적으로 다음과 같은 방식으로 gradient boosting을 수행하고 정확도와 수행시간을 파악(Radom forest보다 좋은 성능을 나타내지만(예측 성능이 더 좋음)) 수행시간이 오래 걸린다는 단점

*rom sklearn.ensemble import GradientBoostingClassifier*

*import time*

*import warnings*

*warnings.filterwarnings('ignore')*

*X\_train, X\_test, y\_train, y\_test = get\_human\_dataset()*

*#gbm 수행 시간 측정*

*start\_time = time.time()*

*gb\_clf = GradientBoostingClassifier(random\_state = 0)*

*gb\_clf.fit(X\_train,y\_train)*

*gb\_pred = gb\_clf.predict(X\_test)*

*gb\_accuracy = accuracy\_score(y\_test,gb\_pred)*

*print('GBM 정확도:{0.4f}'.format(gb\_accuracy))*

*print('GBM 수행시간:{0.4f}'.format(time.time() - start\_time))*

**<hyper parameter>**

좋은 성능, 정확도를 높이기 위해 GridSearchCV를 기반으로 hyper parameter 최적화

대략적으로 다음과 같은 방식으로 hyper parameter 수행

*from sklearn.model\_selection import GridSearchCV*

*params = {*

*'n\_estimators':[100,50],*

*'learning\_rate':[0.05,0.1]*

*}*

*grid\_cv = GridSearchCV(gb\_clf, param\_grid = params, cv = 2, verbose = 1)*

*grid\_cv.fit(X\_train,y\_train)*

*print('최적 하이퍼 파라미터:\n',grid\_cv.params)*

*print('최고 예측 정확도:{0.4f}'.format(grid\_cv.best\_score\_))*

hyper parameter 관련 개념

1. loss : default 값으로 deviance를 지정
2. learnig\_rate : gradient boosting이 학습을 진행할 때마다 적용하는 학습률 default는 0.1이고 0-1 사이의 값을 지정
3. n\_estimators : weak leaner의 개수, default는 100
4. subsample : weak learner가 학습에 사용하는 데이터 샘플링 비율 default는 1