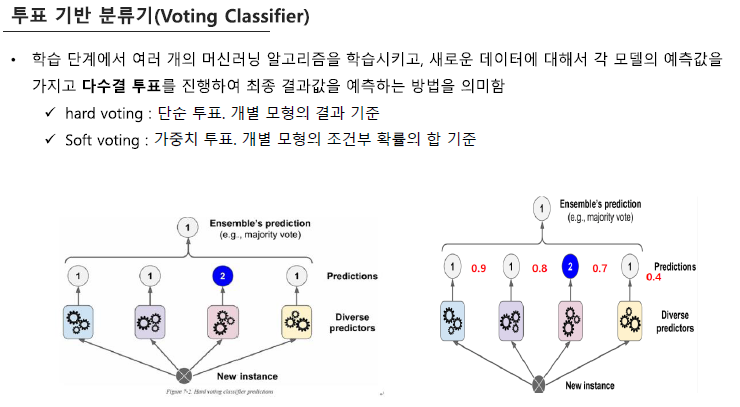
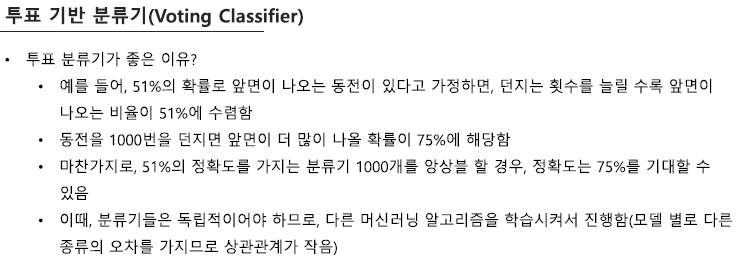


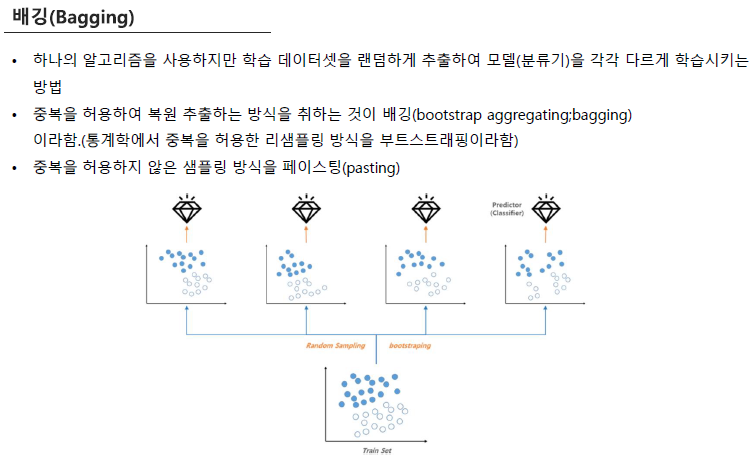
* 주로 성능 향상을 위해 사용함

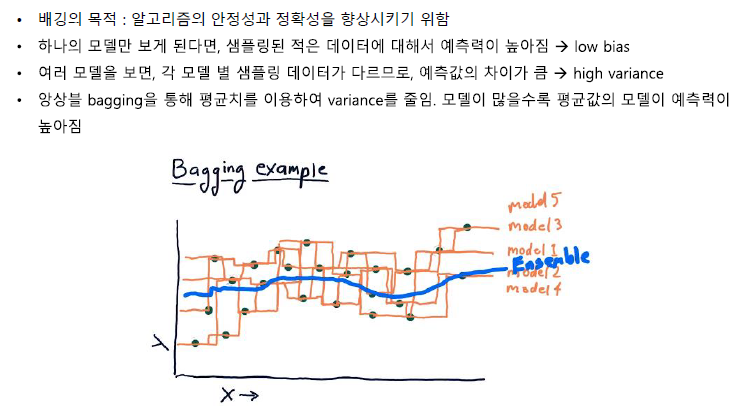


Hard voting: 나온 결과 값 자체를 가지고 결과를 도출하는 것

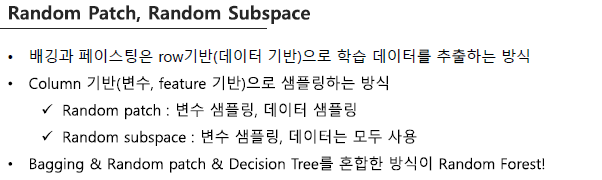
Soft voting: 각 모델이 어떠한 클래스로 분류하면 분류하는 확률이 나오는데 확률 값으로 가중치를 구하고 그에 따른 결과를 도출하는 것





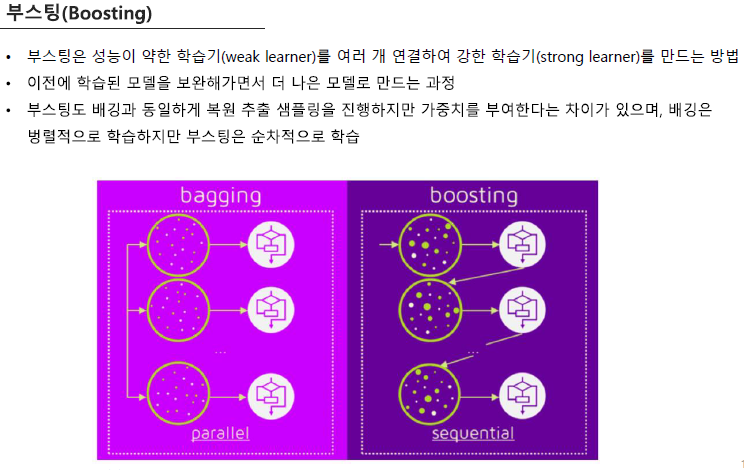


배깅은 모델의 variance를 줄인다.



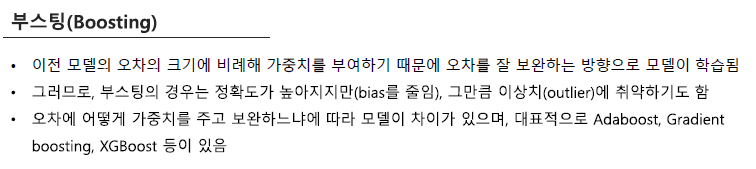
배깅은 샘플링 데이터를 다시 넣어서 사용(복원)

배깅 + 랜덤패치 + DT 🡪 Random Forest



배깅은 각각의 모델은 학습과정이 parallel하게 학습을 시키므로 서로 영향을 주지 않는다.

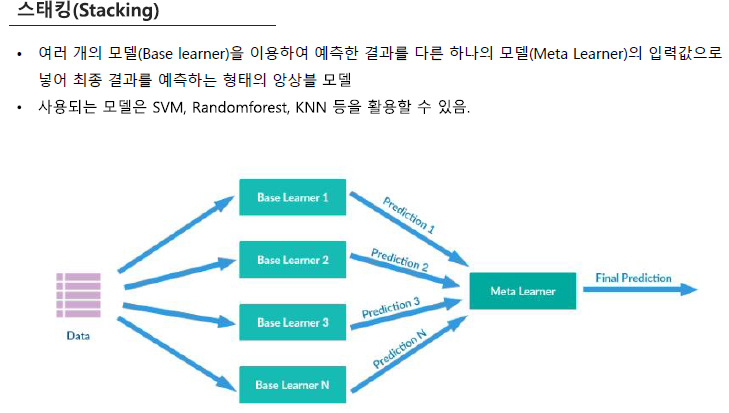
부스팅은 sequential하게 이루어지므로 이전에 나왔던 모델의 결과와 예측 값을 기준으로 더 잘하기 위해서 다음 모델을 구성



XGBoost는 정규화를 사용하는 모델(최근 성능이 좋음)

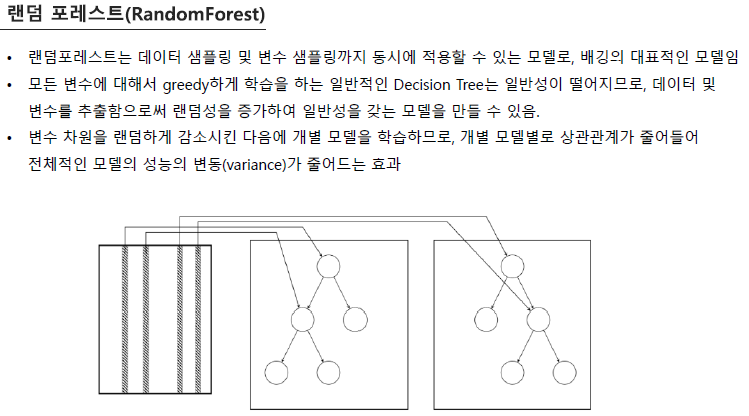
배깅, 부스팅 차이점 – 배깅은 variance를 줄이고 부스팅은 bias를 줄인다.

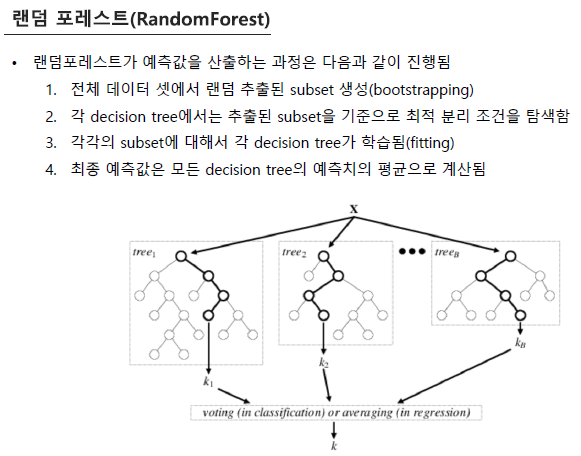
Outlier – 전체 패턴에서 벗어난 데이터

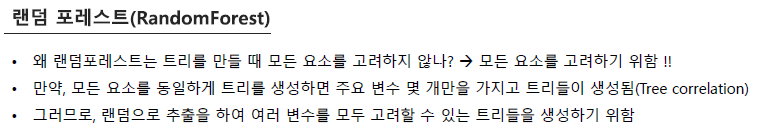


N개의 모델이 만들어지고 예측 값이 각각 나올 때, 배깅은 평균 값을 냈지만 스태킹은 평균 값을 Meta Learner을 더 넣어서 예측 값을 도출(한번 더 모델링을 하는 것)

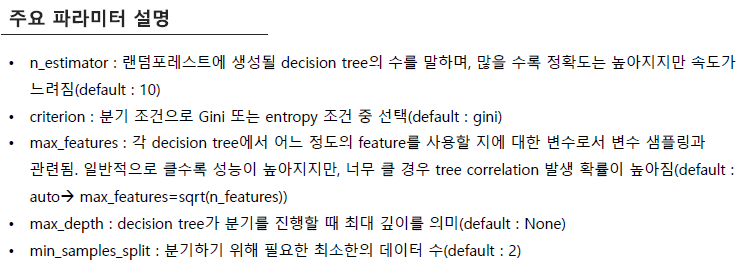
특이점 - 각 모델의 예측 값과 별도로 input 데이터를 넣을 수 있다.

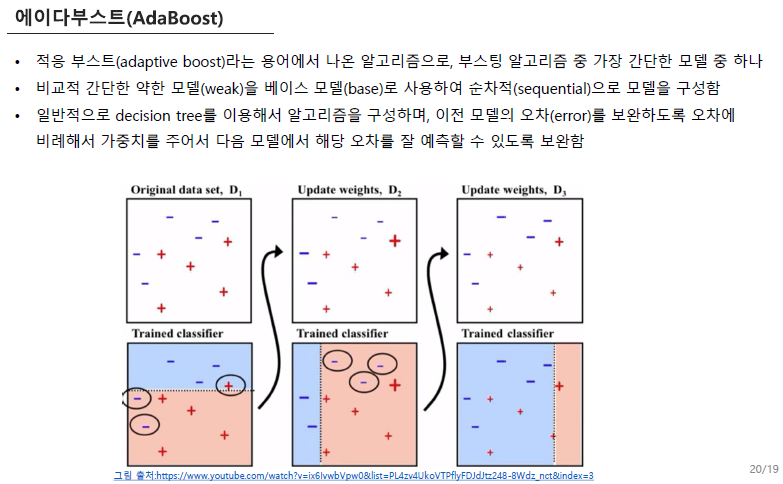




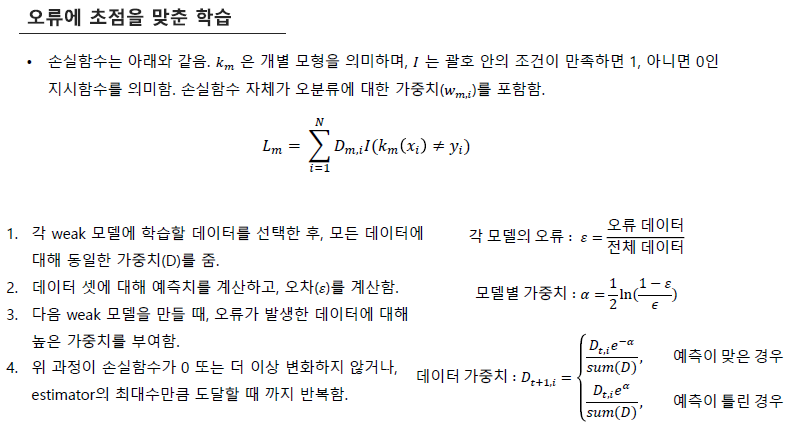


만약에 모든 변수와 모든 데이터를 활용할 경우에는 주요한 몇몇 변수에 의해 좌지우지되는 트리가 만들어지는데, 트리가 만들어지는 구조가 거의 비슷해진다. 이렇게 될 경우, 여러 개를 만들더라도 고려되는 것은 주요한 몇몇 변수 밖에 없다. 하지만 랜덤 샘플링을 통해서 트리를 만들 때, 최대한 많은 변수들을 고려할 수 있다.





학습을 할 경우 잘못 분류된 곳에 가중치를 두어 이전에 잘 맞추는 곳을 잘 맞추는 것보다 이전에 잘 못 맞추는 것을 잘 맞추는 것을 중점으로 잡음



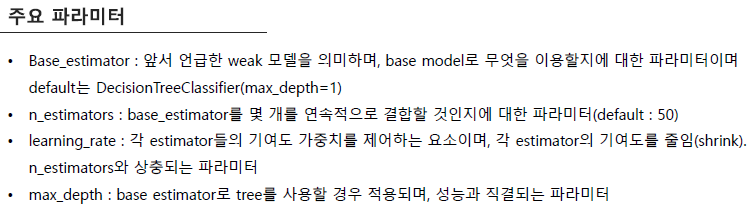
Dt,i - t번째 모델에서 i번째 데이터의 가중치

Yi – i번째 데이터의 실제 값

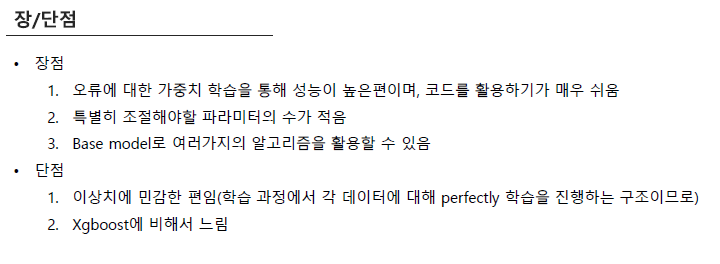
손실함수를 최소화하는 방향을 찾는 것

km(xi) – 예측한 데이터

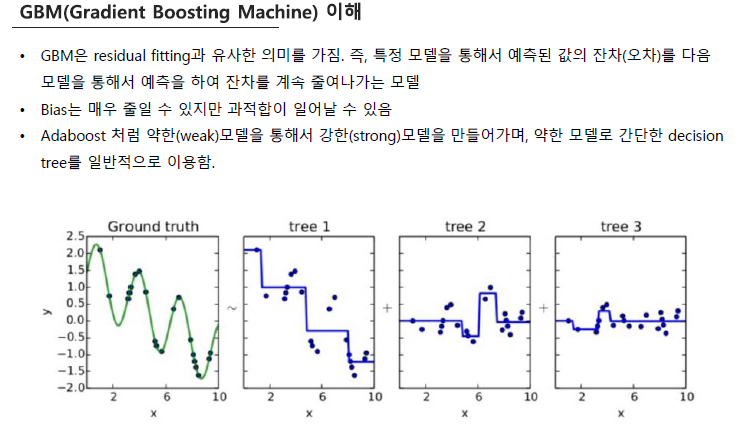
Dm,i - m번째 모델에서 i번째 데이터가 예측 값 km(xi)가 실제 값 yi와 다를 때, 1로 본다

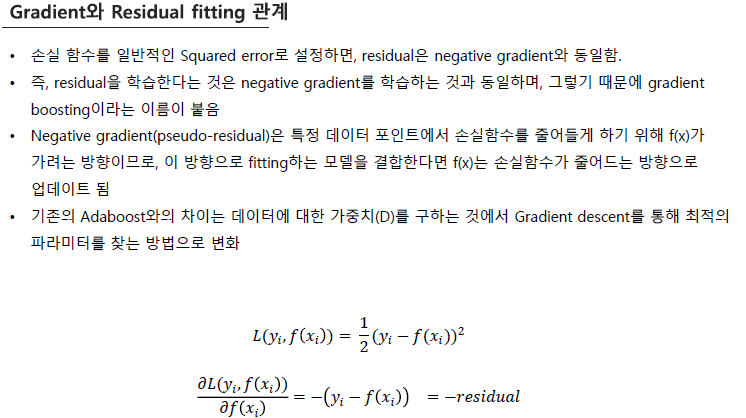


Estimator 개수가 늘어날수록 learning rate가 줄어든다.



2~3번 장점이 주요 장점

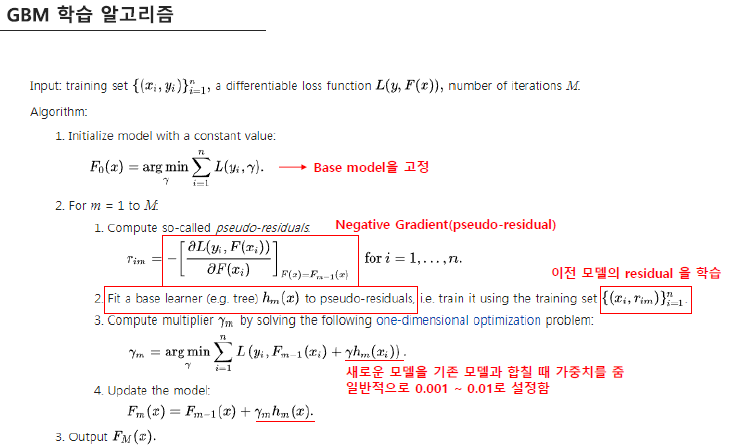




Negative gradient는 함수가 최소화하는 방향을 의미하는 것 🡪 손실함수를 줄어들게 하는 방향

Gradient decent는 -방향으로 점차 학습을 해 나감

Adaboost는 가중치를 부여 vs GBM은 장차(오차)를 구해 gradient descent를 통해 최적 파라미터를 찾음

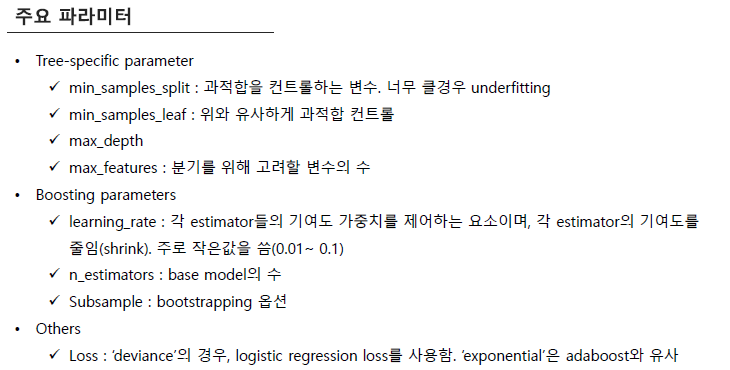


각 모델에서 학습해서 만들어진 장차를 계속 더해서 결과를 도출

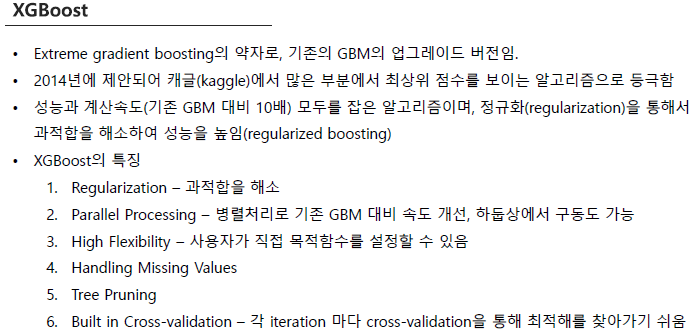
Loss에 미분을 해서 gradient를 구하고 residual을 구함

rim – 장차

loss함수를 최소화할 수 있는 감마를 찾는 것



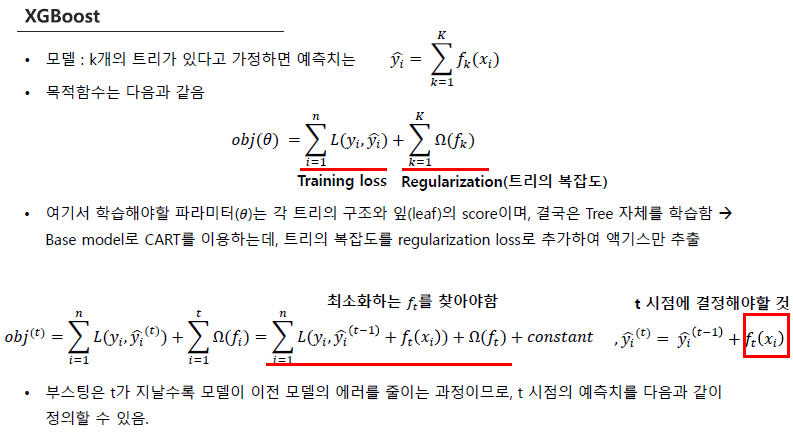
Bootstrapping – 데이터를 얼마만큼 할 것인지



캐글 – 데이터들이 오픈 되어 있고 분석되어 있는 사이트

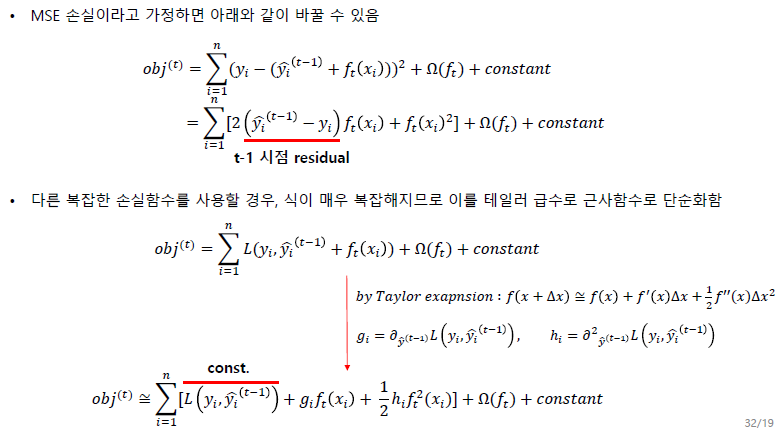
XGBoost는 최상위권의 모델들에 분포

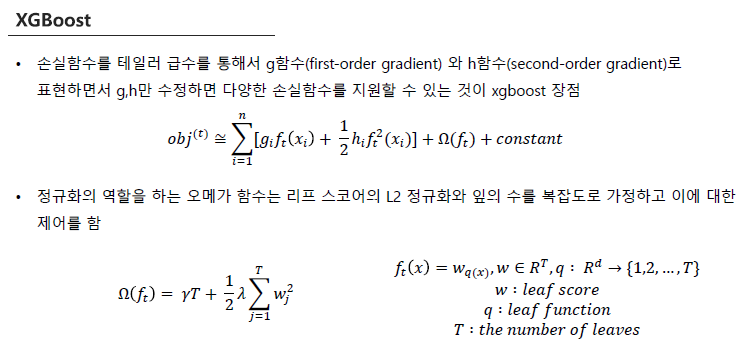
GBM보다 성능, 속도 측면에서 업그레이드 시킴, XGBoost에서 정규화(Regularization)를 통해 테스트 성능을 높임, 사용자가 수식을 직접 이용할 수 있다.



목적 함수는 기존에 GBM이 가지고 있는 손실 함수에 Omega를 추가(정규화에 관련된 요소)

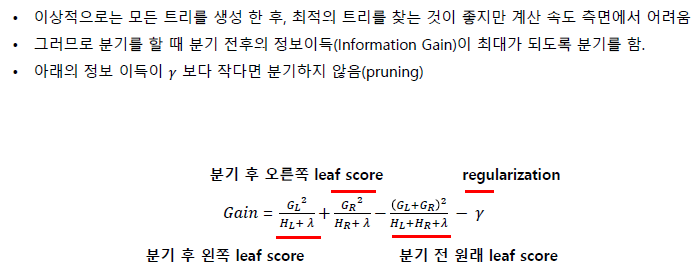
정규화는 Tree에 복잡도를 해소할 요소





감마와 람다의 차이점은 leaf score에 중점을 두는지 leaf의 수에 중점을 두는지 차이임

W^2은 L2정규화를 따른다 🡪 즉 람다(λ)는 L2정규화를 따름



Leaf score는 불순도라고 생각하면 됨

분기 전, 후의 스코어가 람다보다 작게 되면 굳이 쪼갤 필요 없다는 것을 의미 🡪 감마(regularization)는 분기를 나누는 기준

감마가 커질수록 분기가 쉽게 되지 않고 모델이 복잡해지지 않는다.

