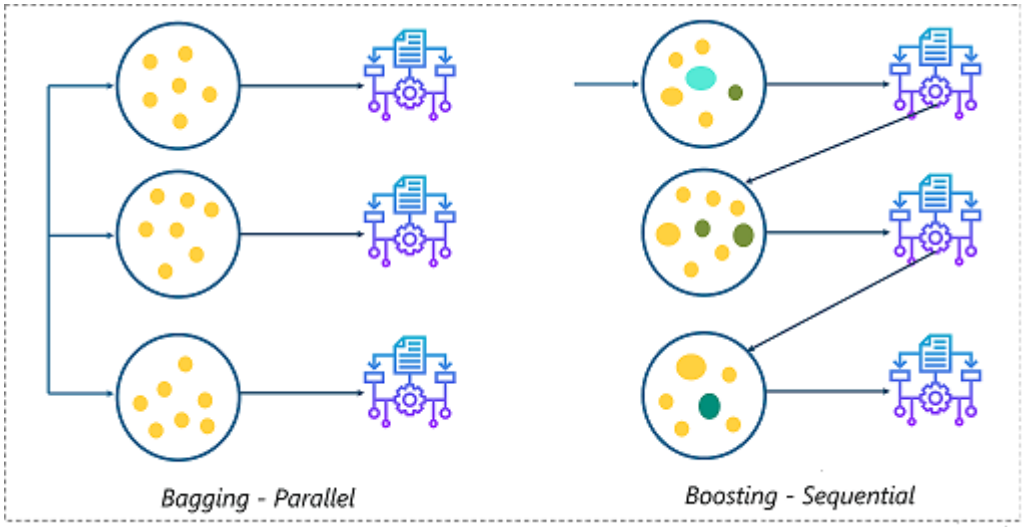
부스팅(Boosting)

머신러닝에서 부스팅은 오차를 줄이기 위해 사용되는 학습 방법. 앙상블 기법 중 하나로 약한 모델 여러개를 결합하여 성능을 높이는 알고리즘.



배깅이 여러 데이터셋으로 나눠 학습한다면 (동시,병렬 학습), 부스팅은 데이터셋 모델이 뒤의 데이터 셋을 정해주고 앞의 모델들을 보완해 나가며 학습.(순차,직렬 학습)

즉, 약한 학습기(weak learner)들을 순차적으로 여러 개 결합하여 강한 예측 모형을 만드는 것

* 약한 학습기: 자체적으로는 성능이 떨어지는 학습 모델. 보통 의사 결정 트리(decision trees)와 같은 간단한 모델이 사용된다.

- 주요 부스팅 알고리즘

- **AdaBoost (Adaptive Boosting):** 가장 널리 알려진 부스팅 알고리즘으로, 관측치들에 가중치를 더하면서 동작을 한다. 분류하기 어려운 Instances에는 가중치를 더하고 이미 잘 분류되어진(다루어진) Instances는 가중치를 덜 한다.

  즉, 약한 학습기(weak learner)의 오류에 가중치를 더하면서 부스팅을 수행하는 알고리즘

AdaBoost의 작동 원리

초기화: 학습 데이터의 모든 샘플에 동일한 가중치를 부여.

반복 학습:

1. 현재 가중치를 사용하여 데이터에 대한 약한 학습기를 훈련시킵니다.
2. 학습기의 성능을 평가하고, 각 샘플에 대한 오류율을 계산.
3. 학습기에 가중치를 할당. 이 때, 오류율이 낮을수록 학습기의 가중치가 더 높아짐.

4. 올바르게 분류되지 않은 샘플의 가중치를 증가, 올바르게 분류된 샘플의 가중치를 감소.

5. 새로운 가중치로 데이터를 업데이트.

모델 결합: 모든 반복 후, 각 학습기의 가중치를 고려하여 최종 모델을 형성.

- **장점**:

파라미터 설정이 간단 (주요설정 약한학습기의 수와 관련), 높은 정확도의 모델 생성

- **단점**:

이상치에 민감, 계산비용(반복적인 과정과 가중치 재조정)

- **Gradient Boosting (GB):** GBM은 AdaBoost처럼 앙상블에 이전까지의 오차를 보정하도록 예측기를 순차적(Sequential)으로 추가. 하지만, AdaBoost처럼 매 반복 마다 샘플의 가중치를 조정하는 대신에 이전 예측기가 만든 잔여 오차(Resudial Error)에 새로운 예측기를 학습시킴.
  - 가중치 업데이트를 경사하강법(Gradient Descent) 기법을 사용하여 최적화된 결과를 얻는 알고리즘
  - 이전 모델의 Residual를 가지고 weak learner를 강화함
  - 즉, Residual를 예측하는 형태의 모델.
  - 과적합(Overfitting) 이슈가 있음.

* 경사하강법: 최적화 문제를 풀기 위한 일반적인 방법, 함수의 값이 낮아지는 방향으로 각 독립변수들의 값을 변형시키면서 함수가 최솟값을 갖도록 하는 독립변수의 값을 탐색 방법 ->손실 함수(또는 비용 함수)의 값이 최소값인 최적의 파라미터(α) 값 찾기

$\theta := \theta - \alpha \cdot \nabla_\theta J(\theta)$ α는 학습률(learning rate) ,$\nabla_\theta J(\theta)$:θ 에서의 손실함수 경사

Ex) 등산에서의 목적지 산 밑-> 계속해서 낮은 지점 찾아 이동

Gradient Boosting 작동 원리

1. 초기 모델 설정: 처음에는 매우 단순한 모델을 사용하여 데이터를 학습(이 모델은 데이터의 패턴을 대략적으로 예측.)

2. 오류 계산: 첫 번째 모델의 예측과 실제 값 사이의 차이(오류)를 계산.

3. 오류 수정을 위한 학습: 다음 단계에서는 이 오류를 줄이기 위한 새로운 모델을 학습시킴. 즉, 첫 번째 모델이 틀린 부분을 보완하도록 두 번째 모델을 특별히 학습시킴.

4. 모델 추가 및 합산: 이 과정을 반복하면서 각 단계에서의 모델을 모두 합산합니다. 즉, 각 모델

은 이전 모델의 오류를 줄이는 데 집중하여 전체 모델의 성능을 점진적으로 향상시킴.

5. 종료 조건: 더 이상 성능 향상이 없거나, 설정한 모델의 수에 도달하면 프로세스를 종료.

- 장점:

높은 정확도: Gradient Boosting은 다른 알고리즘과 비교했을 때 매우 경쟁력 있는 정확도를 제공.

유연성: 다양한 손실 함수를 사용할 수 있어, 회귀부터 분류까지 다양한 유형의 문제에 적용 가능

- 단점:

파라미터에 민감: 잘못된 파라미터 설정은 모델의 성능을 크게 저하

학습 시간: 데이터가 많고, 모델이 복잡할 경우 학습 시간이 길어질 수 있음.

- **XGBoost (Extreme Gradient Boosting):** Gradient Boosting을 기반으로 한 알고리즘으로, 계산 속도와 모델 성능을 향상시키는 다양한 기술이 추가. 또한 대규모 데이터셋 처리에 효율적이다. -> GBM의 단점을 보완하기 위해 XGBoost 모델이 나옴.

  기본 원리: 여러 개의 결정트리를 조합하여 사용하는 기법. 이전 트리의 오류를 차례로 보정해 나가면서 새로운 트리를 추가하는 방식으로 작동. 각 트리는 이전 트리들의 예측을 개선하는 데 기여하며, 모든 트리의 예측 결과는 합쳐짐
  - GBM 보다 빠른 속도를 가짐.
  - 과적합(Overfitting) 방지를 위한 규제(Regularization)가 있음.
  - CART(Classification And Regression Tree) 기반

  즉, 분류(Classification)와 회귀(Regression) 둘 다 가능.

  - GBM과 마찬가지로 가중치 업데이트를 경사하강법(Gradient Descent) 기법을 사용

.

XGBoost 작동 원리

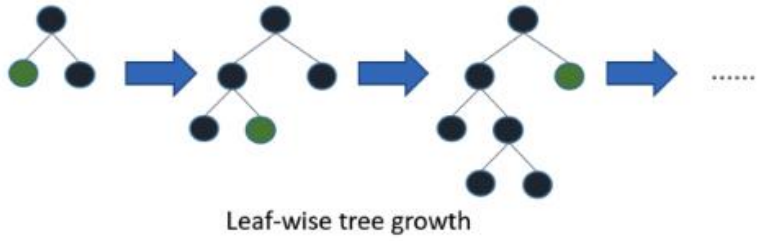
첫 번째 트리 생성-> (실제값, 예측값) 잔차 계산-> 두 번째 트리 생성 -> 잔차 계산 -> 추가 트리 생성 ->지정된 횟수 반복 or 잔차가 더 이상 안줄어듬-> 모델 결합

- **LightGBM:** XGBoost의 장점은 계승하고 단점은 보완하는 방식으로 개발된 부스팅 모델, 더 빠른 학습 속도와 낮은 메모리 사용을 목표로 한다. 큰 데이터셋과 높은 차원의 데이터에서 더욱 효과적이다.

  기본원리: 그래디언트 부스팅 기법과 유사하게 여러 결정 트리를 이용해 순차적으로 학습하는 앙상블 기법을 사용.

  리프 중심 트리 분할(Leaf Wise) 방식을 사용 - 최대 손실 값(Max data loss)을 가지는

리프 노트를 지속적으로 분할하면서 트리의 깊이가 깊어지고 비대칭적인 트리가 생성



Light GBM 작동 방식

- 장점:

속도와 효율성: 훈련 시간과 메모리 사용량이 매우 적다.

대규모 데이터셋 처리: 매우 큰 데이터셋도 효과적으로 처리 가능

고성능: 다양한 데이터셋에서 우수한 예측 성능을 보여줌.

- 단점:

과적합 : 작은 데이터셋이나 불균형한 데이터셋에서 과적합이 발생할 수 있음.

파라미터 튜닝: 최적의 성능을 얻기 위해 다양한 파라미터를 적절히 조정.

◆ **CatBoost(Categorical Boosting):** 범주형 데이터를 자동으로 변환하며, 이로 인한 오버피팅(overfitting- 과적합)을 최소화할 수 있는 Gradient Boosting 기반의 알고리즘. 사용이 쉽고, 다양한 데이터 유형에 강점을 가진다.

- Level-wise Tree: 균형트리방식 사용하여 학습 진행
- Order Boosting: 데이터 셋의 이부를 뽑아 잔차를 계산하고 모델을 만듬
- 자동 범주형 처리방법: Cat는 범주형(Category) 변수를 전처리 할 필요 없이 자동으로 처리하고 최적화.
- 결손값 처리: 결손값(데이터에 값이 없는 경우)을 자동으로 처리, 사용자가 별도로 처리하지 않아도 됨.

-장점:

범주형 데이터 처리: CatBoost는 범주형 데이터 처리에서 뛰어난 성능.

속도와 효율성: 학습 및 예측 과정이 매우 빠르며, 대용량 데이터셋에 적합.

과적합 방지: 고유의 부스팅 기법과 데이터 처리 방식으로 과적합을 효과적으로 방지.

사용 용이성: 복잡한 데이터 전처리 과정 없이도 효과적으로 모델을 학습 가능.

-단점:

파라미터 튜닝: 최적의 성능을 얻기 위해서는 다양한 파라미터를 조절할 필요

리소스 요구량: 최적의 성능을 달성하기 위해 상대적으로 많은 메모리와 계산 리소스를 요구.