CATBOOST

기존의 부스팅 기법

1. 실제 값들의 평균과 실제 값의 차이인 잔차(Residual)를 구한다.
2. 데이터로 이 잔차들을 학습하는 모델을 만든다.
3. 만든 모델로 예측하여, 예측 값에 Learning_rate 를 곱해 실제 예측 값(평균 + 잔차예측 값 *lr) 을 업데이트 한다.
4. 1~3 반복

문제점

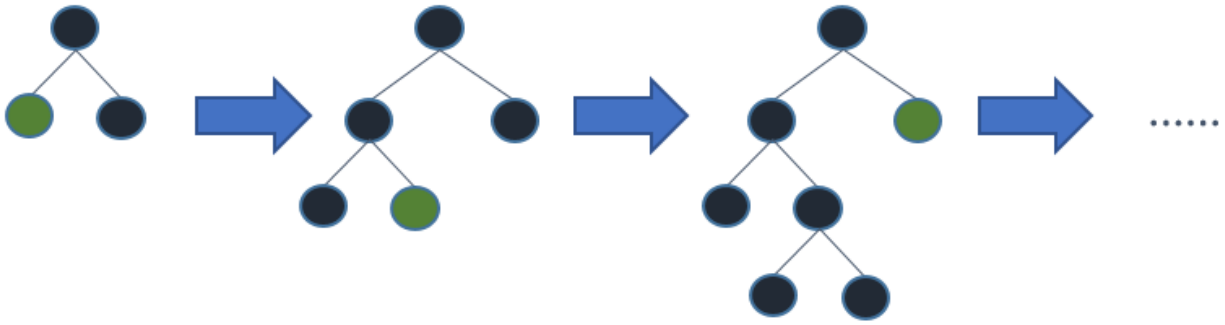
1.느린 학습 속도 (순차적으로 학습하기 때문)
2.과적합 - 잔차를 줄여가는 기법이기 때문에 당연한 결과...

CATBOOST 특징

1.Level-Wise-Tree (기존 GBM은 Leaf-Wise)

***Level-Wise-Tree**

***Leaf-Wise**



Leaf-wise tree growth

2.Orderd Boosting

GBM - 모든 훈련데이터를 대상으로 잔차를 계산함

CatBoost - 일부만 잔차 계산을 진행

time	datapoint	class label
12:00	x1	10
12:01	x2	12
12:02	x3	9
12:03	x4	4
12:04	x5	52
12:05	x6	22
12:06	x7	33
12:07	x8	34
12:08	x9	32
12:09	x10	12

```
1.  먼저 x1 의 잔차만 계산하고, 이를 기반으로 모델을 만든다. 그리고 x2 의 잔차를 이 모델로 예측한다.
2.  x1, x2 의 잔차를 가지고 모델을 만든다. 이를 기반으로 x3, x4 의 잔차를 모델로 예측한다.
3.  x1, x2, x3, x4 를 가지고 모델을 만든다. 이를 기반으로 x5, x6, z7, x8 의 잔차를 모델로 예측한다.
4.  ... 반복
```

3.Random Permutation

2번의 과정을 순서를 뒤섞어 매번 다른 순서로 잔차를 계산하게 만드는 방식 (과적합 방지)

4.Ordered Target Encoding

예시 데이터

time	feature 1	class_labels (max_tempetarure on that day
sunday	sunny	35
monday	sunny	32
tues	couldy	15
wed	cloudy	14
thurs	mostly_cloudy	10
fri	cloudy	20
sat	cloudy	25

```
x=['time','feature1']
y=['class_labels']

1.기존의 mean_encoding 기법
 데이터 누수가 발생하는 문제 - 과적합의 주요 원인
 cloudy = (15+14+20+25)/4 = 18.5

2.Catboost Ordered Target Encoding 기법
 Friday 에는, cloudy = (15+14)/2 = 15.5 로 인코딩 된다.
 Saturday 에는, cloudy = (15+14+20)/3 = 16.3 로 인코딩 된다.
 *이처럼 현재의 데이터 값을 제외한 과거의 데이터만의 평균을 산출하는 방법이다.
```

5.Categorical Feauture Combinations

information gain이 같은 feature들을 하나로 묶는 방법 (변수 선택 부담 감소)

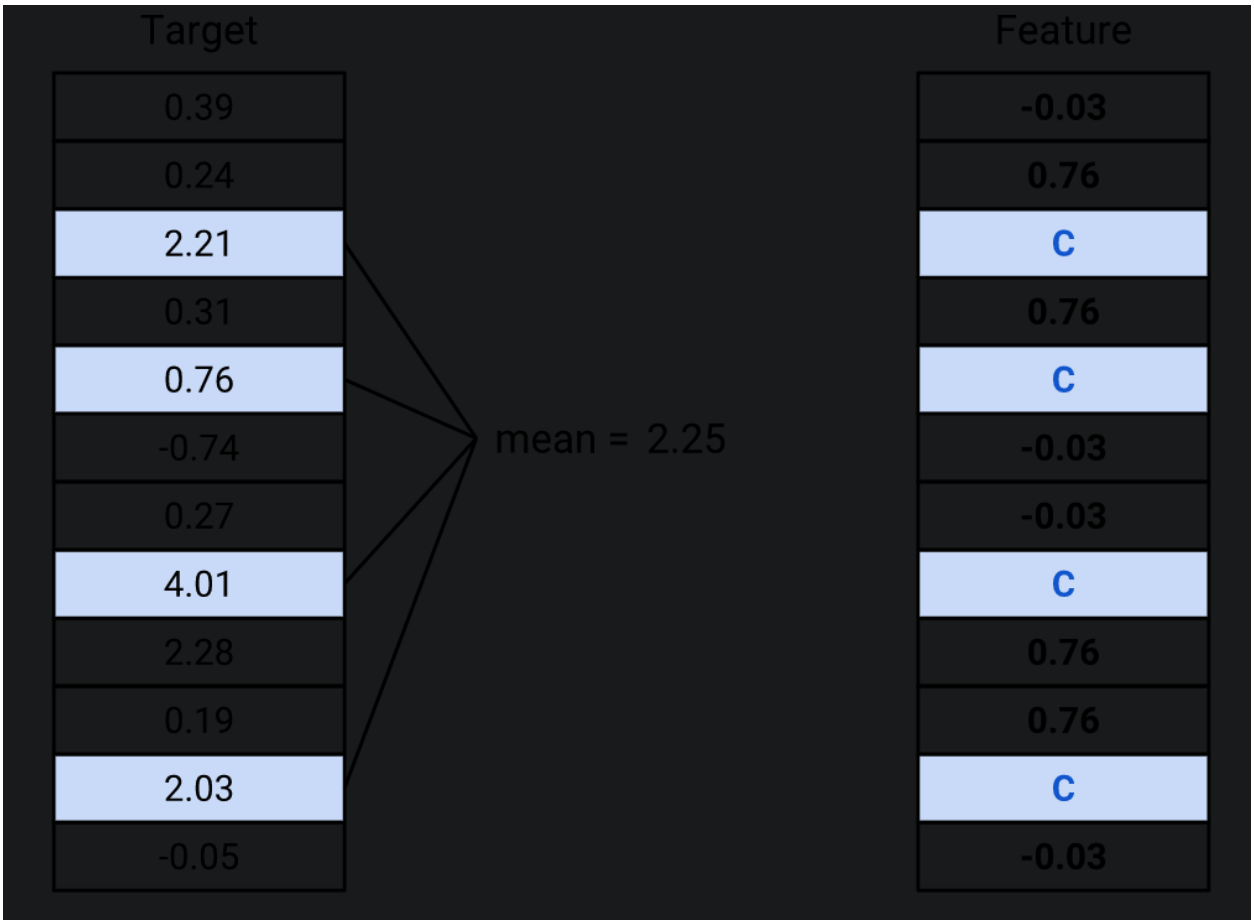
6.One-hot Encoding

집합의 범위 ≤ 3(기본값) - One-Hot Encoding 적용

집합의 범위 > 3 - Target Encoding 적용

***Target Encoding과 one-hot encoding의 단점을 이해해야 함.**

Target Encoding - 각 카테고리를 Target의 평균값으로 대체하는 방법



One-Hot Encoding의 단점

1.One-Hot Encoding의 경우 낮은 집합의 범위(Cardinality)를 가지는 경우 효율이 좋지만

반대로 범위가 너무 큰 경우에는 과도하게 차원이 늘어나 모델링을 함에 있어 큰 단점으로 작용함.

2.모델이 카테고리를 숫자로 인식해 순서를 매기는 위험이 존재함.

```
One-Hot Encoding 집합의 범위는 하이퍼 파라미터를 통해 재설정이 가능
기본값 = 3
catboost = CatBoostClassifier(one_hot_max_size=n)
```

7.Optimized Parameter tuning

기본적으로 하이퍼 파라미터의 최적화가 잘 되어있어 크게 튜닝을 하지 않아도 됨

*XGBoost & GBM의 경우 하이퍼 파라미터에 매우 민감하다는 단점이 존재함.

8.GPU

GPU를 지원하는 환경인 경우 GPU모드 설정이 가능하다.

```
from catboost import CatBoostClassifier
catboost = CatBoostClassifier(task_type='GPU')
```

CatBoost 단점

1.고차원의 데이터 셋의 경우 성능이 급격하게 떨어짐

2.데이터의 대부분이 수치형인 경우 LGMB보다 학습 속도가 느리다.

참고 사이트

1.Target Encoding

https://brendanhasz.github.io/2019/03/04/target-encoding

2.CatBoost

https://hanishrohit.medium.com/whats-so-special-about-catboost-335d64d754ae

https://gentlej90.tistory.com/100

033