CatBoost

CatBoost

https://www.notion.so/CatBoost-84c848de66c34ddcb547ca88816447df?pvs=4

Background

Category & Boosting

> 💡 **Boosting**
>
> `to improve`라는 뜻
>
> Weak learner를 여러 번 사용하여 성능을 높이는 방법
>
> 분석 절차
>
> 1. Weak learner를 이용해 분류한 후, n개의 학습 데이터 중
>   - 분류가 올바르게 된 학습 데이터 → 가중치 감소
>   - 오분류 된 학습 데이터 → 가중치 증가
>
>   ⇒ 오분류된 데이터가 추출될 확률 증가!
> 2. 위를 M번 반복
>
>   이 때, L_i를 i번째 learner라고 하면, $L = \Sigma L_i$
>
> ⇒ `addictive model` : 비모수 회귀 (함수의 형태를 가정하지 않는 회귀 모형)

이전의 boosting 모형

`LightGBM` & `XGBoost`

1. 모든 데이터를 이용하여 모형을 학습한 후, 적용한 데이터로부터 잔차를 구함

   → 모든 관측치에 하나의 동일한 모형으로 잔차를 구함
2. 잔차를 목적 변수로 하여, 동일한 특성 변수를 재적합

   → 동일한 특성 변수를 반복적으로 사용

3. 위의 절차를 M번 반복

⇒ 과대 적합(overfitting)이 자주 발생!

CatBoost 특징

대부분 과대 적합을 방지하기 위한 방법

1. **Level-wise tree**
2. **Ordered Boosting**
   - Existing boosing models는 모든 train data를 대상으로 residual을 계산하지만, CatBoost는 일부만 가지고 잔차를 계산하여 모델링
   - `단계적`으로 데이터를 늘려가면서 모델링 진행 → 모델링할 때마다 달라짐
     1. 먼저 x_1의 잔차만 계산한 후, 이를 기반으로 모델을 만들고, x_2의 잔차를 이 모델로 예측
     2. x_1, x_2의 잔차로 모델을 만들고, x_3, x_4의 잔차를 모델로 예측
     3. x_1, x_2, x_3, x_4의 잔차로 모델을 만들고, x_5, x_6, x_7, x_8의 잔차를 모델로 예측
     4. ... 반복
3. **Random Permutation**
   - Ordered Boosting 과정에서 데이터 순서를 섞어주지 않으면 매번 같은 순서대로 잔차를 예측하는 모델이 만들어질 가능성 높음
   - 매 셔플링마다 임의로 정해진 순서로 데이터가 `랜덤 정렬`
   - 과대 적합 방지를 위한 시도
4. **Ordered Target Encoding**
   - `Target Encoding`, `Mean Encoding`, `Response Encoding` 등으로도 불림
   - 범주형 변수를 numeric 형태로 encoding하는 방법 중 비교적 가장 최근에 나온 기법
     - $Cloudy = (15 + 14 + 20 + 25)/4 = 18.5$

time	feature	class
SUN	Sunny	35
MON	Sunny	32

time	feature	class
TUE	Cloudy	15
WED	Cloudy	14
THU	Mostly_Cloudy	10
FRI	Cloudy	20
SAT	Cloudy	25

    - 해당 값을 가진 데이터들의 class 값의 평균으로 인코딩 (Mean Encoding)

  ⇒ 예측해야 하는 값이 train dataset의 feature로 들어가 버리는 상황! (`Data Leakage`), 과대 적합을 일으키는 주 원인

- **해결책**: 이전 데이터들의 인코딩 값 사용, 현재 target의 값이 아닌 이전 target 값을 사용하여 data leakage 회피

$$FRI : Cloudy = (15 + 14)/2 = 15.5$$

$$SAT : Cloudy = (15 + 14 + 20)/3 = 16.3$$

5. **Categorical Feature Combinations**
   - `Information Gain`이 같은 두 features를 하나로 만들어 데이터 전처리에 있어 `feature selection` 역할
6. **One-hot Encoding**
   - 범주형 변수를 항상 Target Encoding하지는 않음
   - 낮은 `Cardinality`를 가지는 변수에 한해서, 기본적으로 `one-hot encoding` 시행
     - 자동으로 처리, 옵션으로 기준 지정 가능
   - Low Cardinality를 가지는 범주형 변수는 one-hot encoding이 효율적
7. **Optimized Parameter Tuning**
   - 기존 Boosting 모델 (XGBoost, LightGBM)은 파라미터 튜닝에 매우 민감하나, CatBoost는 `기본 파라미터`가 기본적으로 `최적화`가 잘 되어 있음
   - 파라미터 튜닝에 신경 쓰지 않아도 됨; 파라미터 튜닝 전후 결과 차이가 크지 않기 때문

> 📌 데이터 대부분이 수치형 변수인 경우, LightGBM보다 속도가 느림
>
> 특성 변수에 범주형이 `많은 경우`에 적합!

기존 모형과 차이점

1. 표본의 잔차를 모두 다른 모형으로부터 구함
2. 특성 변수가 범주형일 때, 일반적으로 사용하는 one-hot encoding 대신 `ordered target statistic`으로 전환하여 사용
   - 범주가 2개인 경우: 자동으로 `one-hot encoding`으로 수량화
   - 범주가 3개 이상인 경우: 별도의 처리 없이 `순서목표통계량`으로 전환

   → option으로 기준 지정 가능

📌 범주형 특성 변수의 실수화를 위한 자동화 기능 때문에 `Category and Boosting`을 뜻하는 `CatBoost`라는 이름이 부여

📌 관측치의 잔차가 관측 별로 다른 모형에서 구해지므로 과대 적합이 잘 일어나지 않음

📌 범주형 특성 변수의 자동 실수화 기능 덕분에 수렴 속도가 빠름

~~SVM만 생각하면 진짜...~~