<chapter 4>

5. GBM (gradient boosting machine)

- 특징

부스팅의 한 종류

부스팅은 약한 결합기를 순차적으로 학습, 예측하면서 잘못 예측한 데이터에 가중치 부여를 통해 오류를 개선해 나가는 학습 방식

에이다 부스트와 유사하지만, 가중치 업데이트를 경사 하강법을 이용하는 것이 큰 차이

오류 값 (실제 값 – 예측 값)을 최소화하는 방향성을 가지고 반복적으로 가중치 값을 업데이트하는 것이 경사 하강법

GBM은 CART 기반의 다른 알고리즘과 마찬가지로 분류, 회귀 둘 다 가능

- 하이퍼 파라미터

- Loss

경사 하강법에서 사용할 비용 함수 지정

- Learning_rate

GBM이 학습을 진행할 때 마다 적용하는 학습률. Weaker learner가 순차적으로 오류 값을 보정해 나가는 데 적용하는 계수. 0~1 사이의 값을 지정할 수 있으며 기본 값은 0.1. 너무 작은 값을 적용하면 업데이트 되는 값이 작아져서 최소 오류 값을 찾아 예측 성능이 높아질 가능성이 높다

- N_estimators

Weaker learner의 객수. Weak learner가 순차적으로 오류를 보정하므로 개수가 많을수록 예측 성능이 일정 수준까지는 좋아질 수 있으나 개수가 많을수록 수행 시간이 오래 걸림. 기본값 100

- subsample

Weak learner가 학습에 사용하는 데이터의 샘플링 비율. 기본값 1

6. XGBoost

- 특징

GBM 기반의 머신러닝. 뛰어난 예측 성능, GBM 대비 빠른 수행 시간, 과적합 규제, 자체 내장된 교차검증, 결손 값 자체 처리 등 다양한 장점을 가지고 있음

- 하이퍼 파라미터

● 일반 파라미터

Booster: gbtree 또는 gblinear 선택. 디폴트는 gbtree

Silent: 디폴트는 0이며 출력 메시지를 나타내고 싶지 않을 경우 1

Nthread: CPU의 실행 스레드 개수를 조정. 디폴트는 CPU의 전체 스레드를 다 사용하는 것

● 부스터 파라미터

Eta: GBM의 학습률과 같은 파라미터

Num_boost_rounds: GBM의 n_estimators와 같은 파라미터

Min_child_weight: 트리에서 추가적으로 가지를 나눌지를 결정하기 위해 필요한 데이터들의 weight 총합

Gamma: 트리의 리프노드를 추가적으로 나눌지를 결정할 최소 손실 감소 값

Max_depth: 트리 기반 알고리즘의 max_depth와 같음. 0을 지정하면 깊이에 제한이 없음

Sub_sample: GBM의 subsample과 동일. 트리가 커져서 과적합 되는 것을 제어하기 위해 데이터를 샘플링하는 비율을 지정함

● 학습 태스크 파라미터

Objective: 최솟값을 가져야 할 손실 함수를 정의

Binary, logistic: 이진 분류할 때 적용

Eval_metric: 검증에 사용되는 함수를 정의

7. LightGBM

- 특징

XGBoost와 마찬가지로 부스팅 계열 알고리즘. XGBoost보다 학습에 걸리는 시간이 훨씬 적음

'light'라는 이미지가 예측 성능이 상대적으로 떨어진다고 생각이 들 수 있으나 예측 성능에는 별다른 차이가 없음

일반 GBM 계열의 트리 분할 방법 (트리의 깊이를 효과적으로 줄이기 위한 균형 트리 분할 방식)과 다르게 리프 중심 트리 분할 방식 (최대 손실 값을 가지는 리프 노드를 지속적으로 분할해 생성)을 사용

- 하이퍼 파라미터

- 주요 파라미터

Num_iteration: 반복 수행하려는 트리의 개수를 지정. 크게 지정할수록 예측 성능이 높아질 수 있으나, 너무 크게 지정하면 오히려 과적합으로 성능이 저하될 수 있음

Learning_rate: 0과 1 사이의 값을 지정하며 부스팅 스텝을 반복적으로 수행할 때 업데이트되는 학습률 값

Max_depth: 트리 기반의 max_depth와 같음

Min_data_in_leaf: 결정 트리의 Min_samples_leaf와 같은 파라미터

Num_leaves: 하나의 트리가 가질 수 있는 최대 리프 개수

Boosting: 부스팅의 트리를 생성하는 알고리즘을 기술. Gbdt는 일반적인 그래디언트 부스팅 결정 트리, rf는 랜덤 포레스트

Bagging_fraction: 트리가 커져서 과적합 되는 것을 제어하기 위해서 데이터를 샘플링하는 비율을 지정

Feature_fraction: 개별 트리를 학습할 때마다 무작위로 선택하는 피처의 비율

Lambda_l2: l2 regulation 제어를 위한 값. 피처 개수가 많을 경우 적용을 검토하며 값이 클수록 과적합 감소 효과가 있음

Lambda_l1: l1 regulation 제어를 위한 값. L2와 마찬가지로 과적합 제어를 위한 것

- 하이퍼 파라미터 튜닝 방안

- Num_leaves는 개별 트리가 가질 수 있는 최대 리프의 개수이자 LightBGM 모델의 복잡도를 제어하는 주요 파라미터. Num_leaves의 개수를 높이면 정확도가 높아지지만, 반대로 트리의 깊이가 깊어지고 모델이 복잡도가 커져 과적합 영향도가 커짐

- Min_data_in_leaf는 과적합을 개선하기 위한 중요한 파라미터. 보통 큰 값으로 설정하면

트리가 깊어지는 것을 방지

- Max_depth는 명시적으로 깊이의 크기를 제한. 앞의 것들과 결합해 과적합을 개선하는 데 사용

10. 분류 실습 – 캐글 신용카드 사기 검출

- 언더 샘플링과 오버 샘플링

데이터 세트에서 이상 레이블을 가지는 데이터 건수가 매우 적을 경우 제대로 다양한 유형을 학습하지 못하면 예측 성능의 문제가 발생할 수 있음

언더 샘플링: 많은 레이블을 가진 데이터 세트를 적은 레이블을 가진 데이터 세트 수준으로 감소시키는 방식. 과도하게 정상 레이블로 학습/예측하는 부작용을 개선할 수 있지만, 너무 많은 정상 레이블 데이터를 감소시켜서 정상 레이블의 경우 제대로 된 학습을 수행할 수 없는 문제가 발생할 수도 있음.

오버 샘플링: 적은 레이블을 가진 데이터 세트를 많은 레이블을 가진 데이터 수준을 증식하여 충분한 데이터를 확보하는 방법. 동일한 데이터를 단순히 증식하는 방법은 과적합이 되기 때문에 의미가 없으므로 원본 데이터의 피처 값들을 아주 약간만 변경하여 증식함.