GBM

- **부스팅 알고리즘** : 여러 개의 약한 학습기를 순차적으로 학습-예측하면서 잘못 예측한 데이터에 가중치 부여를 통해 오류를 개선해 나가면서 학습하는 방식
- 대표적 구현 : AdaBoost, 그래디언트 부스트

에이다 부스트

- 데이터에 가중치를 부여하면서 부스팅을 수행하는 알고리즘

그래디언트 부스트

- 에이다 부스트와 유사, but 가중치 업데이트를 **경사 하강법**을 이용해 수행.
- **경사 하강법**
  - 오류식을 최소화하는 방향성을 가지고 반복적으로 가중치 값을 업데이트하는 것
  - 반복 수행을 통해 오류를 최소화할 수 있도록 가중치의 업데이트 값을 도출하는 기법

XGBoost

XGBoost 개요

- 트리 기반의 앙상블 학습에서 가장 각광받고 있는 알고리즘

항목	설명
뛰어난 예측 성능	분류와 회귀 영역에서 뛰어난 예측 성능 발휘
GBM 대비 빠른 수행 시간	병렬 수행 등 다양한 기능으로 GBM에 비해 빠른 수행 성능 보장
과적합 규제	과적함 규제 기능 보유 -> 과적합에 좀 더 강한 내구성
Tree pruning	더 이상 긍정 이득이 없는 분할을 가지치기 해서 분할 수를 줄임
자체 내장된 교차 검증	반복 수행시마다 내부적으로 교차검증을 수행해 최적화된 반복 수행 횟수를 가짐
결손값 자체 처리	결손값 자체 처리 기능 보유

파이썬 래퍼 vs 싸이킷런 래퍼

파이썬 래퍼

- 파이썬 패키지
- 자체적으로 교차 검증, 성능 평가, 피처 중요도 등의 시각화 기능 보유

- **DMatrix** 생성 : XGBoost만의 전용 데이터 세트

싸이킷런 래퍼

- 사이킷런의 다른 유틸리티 그대로 사용 가능
- 하이퍼 파라미터에 약간의 차이 존재
  - eta -> learning_rate
  - sub_sample -> subsample
  - lambda -> reg_lambda
  - alpha -> reg_alpha

LightGBM

- XGBoost보다 학습에 걸리는 시간이 훨씬 적음
- 메모리 사용량도 상대적으로 적음
- 예측 성능도 XGBoost와 큰 차이 x.
- 적은 데이터 세트에 적용할 경우 과적합 발생하기 쉬움.
- 리프 중심 트리 분할 사용
- 카테고리형 피의 자동 변환과 최적 분할

HyperOpt를 이용한 하이퍼 파라미터 튜닝

- Grid Search 방식으로 XGBoost나 LightGBM을 이용해 하이퍼 파라미터 최적화 -> 많은 시간이 소모될 수 있음. -> 베이지안 최적화 사용

베이지안 최적화 개요

- **베이지안 최적화** : 블랙 박스 형태의 함수에서 최대 또는 최소 함수 반환값을 만드는 최적 입력값을 가능한 적은 시도를 통해 빠르고 효과적으로 찾아주는 방식
- 대체 모델과 획득 함수로 이루어짐.
  - **대체 모델** : 획득 함수로부터 최적 함수를 예측할 수 있는 입력값을 추천받은 뒤 이를 기반으로 최적 함수 모델을 개선해 나감
  - **획득 함수** : 개선된 대체 모델을 기반으로 최적 입력값을 계산

HyperOpt 사용법

- 사용 로직
  - a. 입력 변수명과 입력값의 검색 공간 설정
  - b. 목적 함수 설정
  - c. 목적 함수의 반환 최솟값을 가지는 최적 입력값 유추

- 1 - **hp 모듈** : 입력 변수명과 입력값의 검색 간 설정
- 2 - **목적 함수** : 변수값과 변수 검색 공간을 가지는 딕셔너리를 인자로 받고, 특정 값을 반환
- 3 - **fmin 함수** : 최적의 입력값을 베이지안 최적화 기법에 기반하여 찾아줌.

HyperOpt를 이용한 XGBoost 하이퍼 파라미터 최적화

- 주의할 점
  - HyperOpt: 입력값과 반환 값이 실수형 -> 하이퍼 파라미터 입력 시 형변환 필요
  - HyperOpt: 최솟값으로 최적화 -> 값이 클 수록 좋은 성능지표일 경우 -1 곱해줘야 함.

스태킹 앙상블

- 스태킹 : 개별 알고리즘으로 예측한 데이터를 기반으로 다시 예측 수행
- 개별 기반 모델과 **최종 메타 모델** 필요
  - 최종 메타 모델 : 개별 기반 모델의 예측 데이터를 학습 데이터로 만들어서 학습
- -> 여러 개별 모델의 예측 데이터를 스태킹 형태로 결합해 최종 메타 모델의 학습용 피처 데이터 세트와 테스트용 피처 데이터 세트를 만드는 것이 핵심

CV 기반의 스태킹

- 과적합을 막기 위해 최종 메타 모델을 위한 데이터 세트를 만들 때 **교차 검증 기반으로 예측된** 결과 데이터 세트를 이용

단계

1. 각 모델별로 원본 학습/테스트 데이터를 예측한 결과 값을 기반으로 메타 모델을 위한 학습용/테스트용 데이터를 생성
2. 앞 단계에서 개별 모델들이 생성한 학습용 데이터를 모두 스태킹 형태로 합쳐서 메타 모델이 학습할 최종 학습용 데이터 세트를 생성. 테스트용 데이터 또한 스태킹 형태로 합쳐서 최종 테스트 데이터 세트 생성.
3. 최종 학습 데이터 세트와 원본 학습 데이터의 레이블을 기반으로 메타 모델이 학습한 뒤, 최종 테스트 데이터 세트를 예측하고, 원본 테스트 데이터의 레이블 데이터를 기반으로 평가.