

05. GBM(Gradient Boosting Machine)

- GBM의 개요 및 실습
 - 부스팅 알고리즘: 여러 개의 약한 학습기를 순차적으로 학습-예측하며 잘못 예측한 데이터에 가중치 부여를 통해 오류 개선하며 학습하는 방식, 대표적인 구현으로 Adaboost와 Gradient 부스트가 있음
 - Gradient 부스트는 경사 하강법을 이용해 가중치 업데이트
 - 예측 성능 뛰어남
 - 오랜 수행 시간, 더 많은 하이퍼 파라미터 튜닝

06. XGBoost(eXtra Gradient Boost)

- 장점
 - 뛰어난 예측 성능
 - GBM 대비 빠른 수행 시간
 - 과적합 규제
 - 나무 가지치기
 - 교차 검증 자체 내장
 - 결손값 자체 처리
- 파이썬 래퍼 XGBoost 하이퍼 파라미터
 - 일반 파라미터
 - 부스터 파라미터 → 대부분의 하이퍼 파라미터가 속함
 - 학습 태스크 파라미터
- 파이썬 래퍼 XGBoost 적용 - 위스콘신 유방암 예측
- 사이킷런 래퍼 XGBoost의 개요 및 적용
 - 사이킷런 기본 Estimator 상속
 - fit(), predict()만으로 학습과 예측 가능
 - 사이킷런의 다른 유틸리티 그대로 사용 가능
 - 분류를 위한 XGBClassifier, 회귀를 위한 XGBRegressor로 나뉨

07. LightGBM

- 개요
 - 빠른 학습과 예측 수행 시간
 - 적은 메모리 사용량
 - 리프 중심 트리 분할 방식 사용으로 예측 오류 손실 최소화
- LightGBM 하이퍼 파라미터
 - 리프 노드가 분할되며 트리 깊이가 깊어지므로 이에 맞는 하이퍼 파라미터 설정 필요
 - num_leaves의 개수를 중심으로 min_child_samples, max_depth 함께 조정하며 튜닝

08. 베이지안 최적화 기반의 HyperOpt를 이용한 하이퍼 파라미터 튜닝

- 개요
 - 목적 함수 식을 제대로 알 수 없을 때 최적 입력값을 적은 시도를 통해 빠르고 효과적으로 찾아주는 방식
 - 새로운 데이터를 입력받았을 때 최적 함수를 예측하는 사후 모델을 개선해나가며 최적 함수 모델 만들기
 - 대체 모델과 획득 함수가 주요 요소
- HyperOpt 사용하기
 - 입력 변수명, 입력값 검색 공간 설정 / 목적 함수 설정 / 최적 입력값 유추 순으로 로직 구성
 - 다른 패키지과 달리 목적 함수 반환 값의 최솟값을 가지는 최적 입력값을 유추한다는 점 유의
- HyperOpt를 이용한 XGBoost 하이퍼 파라미터 최적화
 - 특정 하이퍼 파라미터 입력 시 실수형으로 형변환해야 함을 유의
 - 최솟값을 반환할 수 있도록 성능 값이 클 수록 좋은 성능 지표일 경우 -1을 곱해야 함

09. 분류 실습 - 캐글 산탄데르 고객 만족 예측

10. 분류 실습 - 캐글 신용카드 사기 검출

- 언더 샘플링과 오버 샘플링의 이해
 - 이상 레이블 데이터 건수가 적어 다양한 유형을 학습하지 못하는 문제 발생
 - 언더 샘플링: 많은 데이터 세트를 적은 데이터 세트 수준으로 감소
 - 오버 샘플링: 적은 데이터 세트를 증식하여 학습을 위한 충분한 데이터 확보

11. 스택킹 앙상블

- 개요
 - 개별적인 여러 알고리즘을 서로 결합해 예측 결과 도출
 - 개별 알고리즘으로 예측한 데이터를 기반으로 다시 예측 수행(메타 모델)
 - 개별적인 기반 모델, 최종 메타 모델 두 종류의 모델을 필요로 함
 - 개별 모델의 예측 데이터를 결합해 최종 메타 모델의 학습용/테스트용 피쳐 데이터 세트를 만듦
- CV 세트 기반의 스택킹
 - 교차 검증 기반으로 예측된 결과 데이터 세트를 이용해 최종 메타 모델을 위한 데이터 세트 만듦
 - 과적합 개선
 - step1: 개별 모델들이 교차 검증으로 학습용/테스트용 스택킹 데이터 생성
 - step2: 생성한 데이터를 기반으로 메타 모델이 학습과 예측 수행