7주차_앙상블 학습과 랜덤 포레스트

1. 앙상블 학습
    - 앙상블 학습 모델은 표 형식으로 저장될 수 있는 정형 데이터 분석에 유용
        - 그렇다고 비정형 데이터에 못 쓰는건 아님. 딥러닝 모델에 적용하면 모델 성능 높이기 가능
    - 편향 : 편향이 크면 과소적합 발생
    - 분산 : 분산이 크면 과대적합 발생

      → 편향과 분산은 trade-off
    - MSE는 $편향^2 + 분산$으로 근사됨
2. 투표식 분류기
    a. 직접 투표
        - 예측값들의 다수로 결정
    b. 간접 투표
        - 예측한 확률값들의 평균값으로 예측값 결정
        - 직접 투표 방식보다 성능이 좋다 (높은 확률에 비중을 두기 때문에)
3. 배깅과 페이스팅
    - 여러 개의 동일한 모델을 하나의 훈련셋의 다양한 부분집합을 대상으로 학습
    - 부분집합 선택할 때 중복 허용 여부에 따라 앙상블 학습 방식이 달라진다
        a. 배깅
            - 중복 허용 샘플링
            - bootstrap aggregation의 줄임말
        b. 페이스팅
            - 중복 미허용 샘플링
    - 예측값
        - 분류 모델 : 예측값 중 최빈값 (직접 투표 방식)
        - 회귀 모델 : 예측값들의 평균값
    - 개별 예측기에 비해 bias는 조금 커지거나 비슷, variance는 줄어듦
    - 배깅이 과대적합 위험성 줄여줌 → 배깅을 기본으로 사용
    - OOB 평가
        - 훈련에 사용하지 않은 모델들의 예측값들의 샘플 이용하여 앙상블 모델 검증하는 기법
4. 랜덤 배치와 랜덤 서브스페이스
    a. BaggingClassifier는 특성에 대한 샘플링 기능 지원
        i. max_features
            - 학습에 사용할 feature 수 지정
            - feature 선택은 무작위
            - max_samples와 유사 기능
        ii. boostrap_features
            - 특성 선택할 때 중복 허용 여부 지정 (기본은 False)
            - Bootstrap과 유사 기능
    b. 랜덤 패치 기법
        - 샘플 + feature 선택

- 각 모델은 일부 샘플 + 일부 feature 조합으로 학습

c. 랜덤 서브스페이스 기법

- feature만 선택
- 각 모델은 전체 샘플에 대해 일부 feature만 사용하여 학습

5. 랜덤 포레스트

a. 배깅/페이스팅 적용한 decision tree의 앙상블을 최적화한 모델

- 분류 용도 : RandomForestClassifier
- 회귀 용도 : RandomForestRegressor

b. 엑스트라 트리

- extremely randomized tree 앙상블이라고 불림
- 무작위로 선택된 일부 특성에 대해 임계값 무작위로 선택 후 그 중에서 최적 선택

c. 특성 중요도

- 해당 특성을 사용한 마디가 평균적으로 불순도 얼마나 감소시키는지 측정
- 불순도를 많이 줄이는 특성 → 중요도 커짐

6. 부스팅

- 이전 학습기의 결과를 바탕으로 예측값의 정확도 조금씩 높여감
  - Gradient Boosting
  - AdaBoost
  - XGBoost
- Gradient Boosting
  - 이전 모델에 의해 생성된 잔차 보정하도록 새로운 예측기 훈련
  - 모델은 결정트리 사용
    - 분류 모델 : GradientBoostingClassifier
    - 회귀 모델 : GradientBoostingRegressor
- learning_rate
  - 훈련된 decision tree 모델 각각이 최종 예측값 계산할 때의 기여도 결정
- 수축 규제
  - 훈련에 사용되는 각 모델의 기여도 줄이는 방식으로 훈련 규제
- 조기 종료
  - `n_iter_no_change` 하이퍼파라미터
- 확률적 그레디언트 부스팅
  - 각 결정트리가 훈련에 사용할 훈련 샘플의 비율 지정하여 학습
  - 훈련 속도 빨라짐
- 히스토그램 그레디언트 부스팅
  - 대용량 데이터셋 이용하여 훈련해야할 때
  - 훈련 샘플의 특성값을 `max_bins` 개의 구간으로 분류
  - 회귀 모델 : HistGradientBoostingRegressor
  - 분류 모델 : HistGradientBoostingRegressor
- XGBoost
  - 그레디언트 부스팅과 차이점?
    - 노드 분할을 통해 낮춰야 하는 비용함수가 다름

        - 불순도 대신 모델 훈련의 목적에 맞는 손실 함수(mse, logloss 등) 사용
        - 생성되는 결정트리의 복잡도가 비용함수 포함됨 → 최종적으로 생성되는 모델에 사용되는 결정트리의 복잡도를 가능한 낮추도록 유도
    - 결측치 포함 데이터 처리 가능
    - 빠른 속도, 확장성