ch7 앙상블, 랜덤 포레스트

앙상블은 예측기가 가능한 독립적일 때 성능 좋음

하나의 예측기를 훈련했을 때보다 편향 비슷하지만 분산 줄어듦 → 오차 비슷하지만 결정경계가 덜 불규칙

투표 기반 분류기 (voting) - 각기 다른 알고리즘을 사용

1. **hard voting** : 각 분류기의 예측을 모아 다수결 투표로 클래스 예측 (다수결)

앙상블에 포함된 가장 좋은 개별 분류기보다 정확도 높을 수 있음 (큰 수의 법칙)

→ 분류기가 완벽하게 독립적이고 오차에 상관관계 없어야 가능

but, 같은 데이터로 훈련시키기 때문에 분류기들이 같은 종류의 오차를 만들기 쉬워 정확도 낮아질 수 있음

2. **soft voting** : 클래스의 확률을 평균해 가장 높은 확률의 클래스로 예측

모든 분류기가 클래스의 확률을 추정할 수 있어야 함

→ SVC : probability=True 지정 필요

hard voting보다 일반적으로 성능 높음

bagging & pasting

같은 알고리즘을 사용해 훈련셋의 서브셋을 무작위로 구성하는 것

1. **bagging** (bootstrap aggregating) : 훈련 세트에서 중복을 허용하여 샘플링하는 방식
2. **pasting** : 중복을 허용하지 않고 샘플링하는 방식

→ 분류일 때는 statistical mode (통계적 최빈값, 가장 많은 예측 결과), 회귀일 때는 평균 계산

⇒ 일반적으로 배깅이 페이스팅보다 성능 좋음 (부트스트래핑 통해 분산 감소, 편향 증가)

oob (out-of-bag) 샘플 : 선택되지 않은 나머지 훈련 샘플

→ 예측기가 훈련되는 동안 oob 샘플을 사용하지 않으므로 별도의 검증 셋 사용하지 않고 oob 샘플로 평가 가능

⇒ 각 예측의 oob 평가를 평균해서 얻음

- bagging의 **feature sampling**

bagging은 feature 샘플링도 지원. 각 예측기는 무작위로 선택한 입력 feature의 일부분으로 훈련됨 (이미지와 같은 고차원 데이터셋을 다룰 때 유용)

1. 랜덤 패치 방식 (random patches method) : feature와 샘플을 모두 샘플링하는 방법
2. 랜덤 서브스페이스 방식 (random subspace method) : 훈련 샘플 모두 사용하고 feature만 샘플링하는 방법

→ 더 다양한 예측기 만들어 편향 증가, 분산 감소시킴

랜덤 포레스트 (random forest)

배깅(페이스팅)을 적용한 decision tree의 앙상블

BaggingClassifier에 DecisionTreeClassifier 넣는 대신 RandomForestClassifier 사용 가능

트리의 노드 분할할 때 최선의 특성을 찾는 것이 아닌 무작위로 선택한 특성 후보 중 최적의 특성을 찾아 랜덤성을 더 주입함 → 편향 증가, 분산 감소시켜 성능 좋아짐

- 엑스트라 트리 (extra-trees, extremely randomized tree ensemble)

각 노드에서 일부 랜덤한 특징을 선택 후, **완전히 랜덤한 값**으로 분할

→ 데이터 샘플링 + 무작위 분할로 더 강한 랜덤성 → 편향 증가, 분산 감소

일반적으로 랜덤 포레스트보다 빠름

- **Feature Importance**

랜덤 포레스트는 특성의 상대적 중요도를 측정하기 쉬움

특정 특성을 사용한 노드가 평균적으로 imputiry를 얼마나 감소시키는지 확인해 특성 중요도를 측정함

Boosting

약한 학습기를 여러 개 연결해 강한 학습기를 만드는 앙상블 방법

1. **AdaBoost** (Adaptive Boosting)

이전 모델이 언더피팅했던 훈련 샘플의 가중치를 높여 학습하기 어려웠던 샘플에 맞춰짐

첫 분류기가 만든 예측에서 잘못 분류된 샘플의 가중치를 상대적으로 높여 두 번째 분류기는 업데이트된 가중치를 사용해 훈련 → 반복

⇒ 병렬화할 수 없음 (배깅, 페이스팅만큼 확장성 높지 않음)

예측기가 정확할수록 가중치가 더 높아짐, 랜덤 예측 정도라면 가중치는 0에 가까움, 랜덤 예측보다 정확도 낮으면 가중치는 음수 됨

→ 가중치 합이 가장 큰 클래스가 예측 결과됨

2. gradient boosting

반복마다 샘플의 가중치를 수정하는 대신 이전 예측기가 만든 잔여 오차에 새로운 예측기 학습

GBRT (gradient boosted regression tree) : 결정 트리 기반 예측기를 사용해 회귀 문제 다루는 그래디언트 부스팅 → GradientBoostingRegressor

파라미터 learning_rate는 각 트리의 기여도 조절, 낮으면 많은 트리 필요하지만 예측 성능 좋아짐 → 축소 (shrinkage)라고 불리는 규제 방법

- stochastic gradient boosting (확률적 그래디언트 부스팅)

특정 비율의 무작위 샘플링된 훈련 샘플로 학습

→ 편향 증가, 분산 감소, 속도 증가

- XGBoost (extreme gradient boosting)

최적화된 그래디언트 부스팅 구현

Stacking (stacked generalization)

모든 예측기의 예측을 취합하는 함수를 사용하는 대신 취합하는 모델을 훈련

일반적으로 hold-out 세트 사용해 blender (meta learner)가 최종 예측 학습

분류기들이 만든 예측값을 입력 특성으로 사용해 새로운 훈련 세트 만들고 블렌더가 새 훈련 세트로 훈련됨 (예측을 가지고 타깃값을 예측하도록 학습)