기계학습_20231031

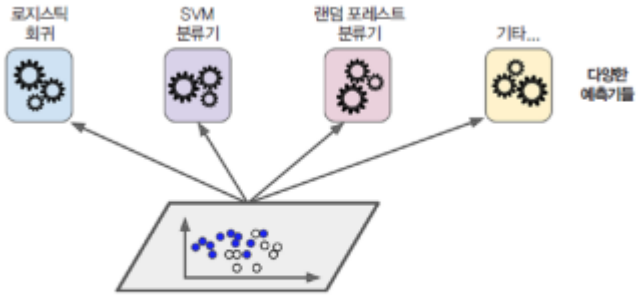
7.0 앙상블 학습과 랜덤포레스트

- 대중의지혜 : 앙상블의 key idea

7.1 투표 기반 분류기

동일 훈련 세트 → 여러 종류 분류기 이용한 앙상블 학습 적용 (앙상블 key idea의 이론버전)

이후 직/간접 투표로 예측

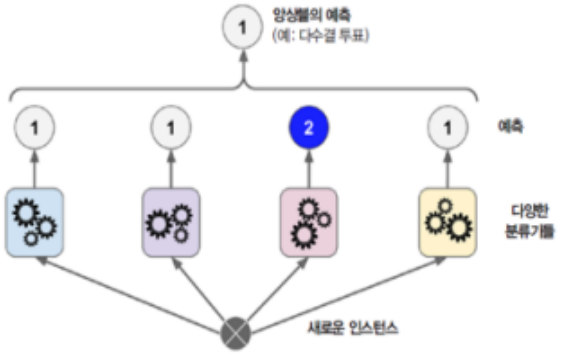


▲ 그림7-1 여러 분류기 훈련시키기

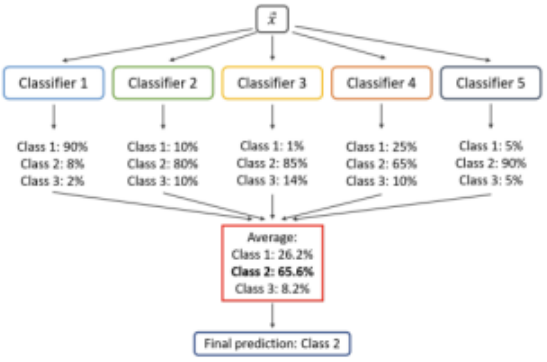
7.1.1 투표 기반 분류기 (직/간접 투표)

직접(hard) : 예측기들의 예측값들을 다수결 투표로 결정

간접(soft) : 예측기들의 예측값들의 평균값으로 결정 → 성능 좀더 좋음



▲ 그림 7-2 직접 투표 분류기의 예측



▲ 그림7-2-1 간접 투표 분류기의 예측

7.1.2(코드) 투표 기반 분류기

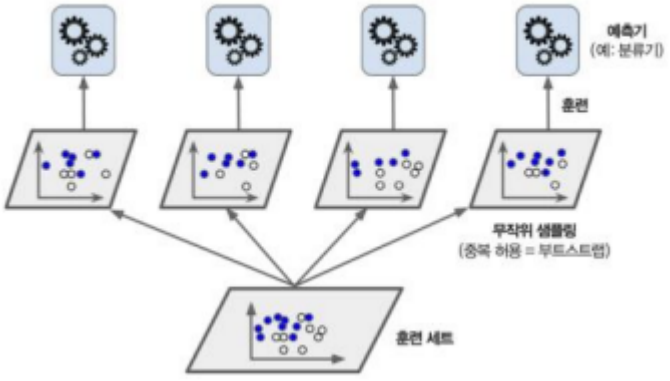
큰 수의 법칙 : 반복 시행하는 횟수가 많거나 표본 커질수록 → 수렴,비교적 정확한 예측

7.2 배깅과 페이스팅

알고리즘은 같은데 훈련세트 다름 : 배깅 페이스팅의 key idea

배깅 : 샘플링 중복 O

페이스팅 : 샘플링 중복 X



7.2.1사이킷런의 배깅과 페이스팅

-배깅 페이스팅

분류모델(BaggingClassifer):최빈값 선택

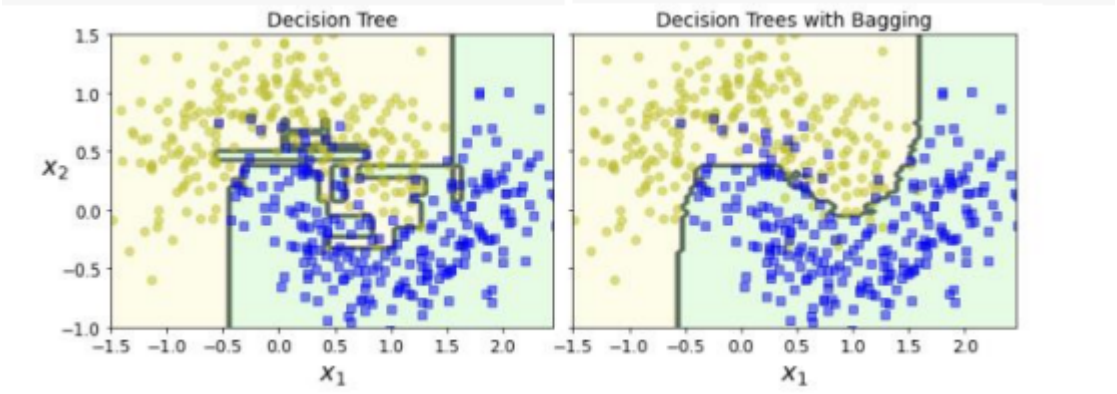
회귀모델(BaggingRegressor):평균값선택

-앙상블학습

편향비슷 분산 줄어듦(분산적음=일반화잘됨/why?=큰수의법칙처럼 많은종류 학습시킬때 일반화가 잘됨)

7.2.1(코드)사이킷런의 배깅과 페이스팅

결과 : 전반적으로 배깅방식 win



7.2.2 oob 평가

배깅 : 어떤 샘플은 여러번 샘플링, 어떤 샘플은 전혀 안쓰임

oob : 선택되지 않은 훈련 샘플 → 앙상블학습에 쓰인 개별 예측기 성능평가(각예측기 oob 평가의 평균)

7.3 랜덤 패치와 랜덤 서브스페이스

랜덤 패치: 훈련샘플,훈련특성 → 중복허용, 임의의 샘플수/특성수 만큼 샘플링해서 학습

랜덤 서브스페이스: 전체훈련세트학습,훈련특성만 임의의 특성수만큼 샘플링해서 학습

7.4 랜덤 포레스트

→ 배깅/페이스팅 방법 적용한 결정트리의 앙상블을 최적화한 모델

알고리즘 : 트리 노드 분할시 최선의 특성을 찾는게 아닌 무작위성 주입

→ 다양한트리, 분산낮춤(편향은 손해)

7.4.1 엑스트라 트리

랜덤포레스트의 노드 분할 방식
- 특성: 무작위 선택
- 특성 임곗값: 무작위로 분할후 최적값 선택
엑스트라 트리(extra-trees)의 노드 분할 방식
특성과 특성 임곗값 모두 무작위 선택 (극단적으로 무작위한 트리의 랜덤 포레스트)
→ 속도 빠름, 분산낮춤(편향증가),

7.4.2 특성 중요도

해당 특성 사용 노드가 평균적으로 불순도를 얼마나 감소시키는지 측정
→ 불순도 줄이면 중요도 커짐
→훈련후 특성마다 중요도의 전체 합이 1이 되도록 결과값을 정규화

7.5 부스팅

성능 약한 학습기 여러 개 연결→ 강한 성능의 학습기를 만드는 앙상블 기법
(but순차적 학습=확장성이 떨어짐)

방식 : 순차적으로 이전 학습기의 결과를 바탕으로 성능을 조금씩 높혀가는 방식
→Key Idea: 앞모델을 보완하며 순차적으로 예측기를 학습

방식 종류 : 에이다부스트 ,그레이디언트 부스팅

7.5.1 에이다부스트

잘못 조정된 가중치를 조정(과소적합한 훈련샘플의 가중치 높임)하여 새로운 예측기 추가

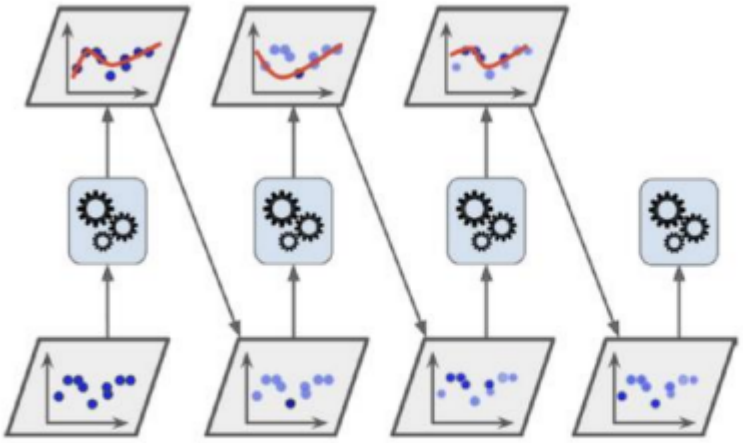
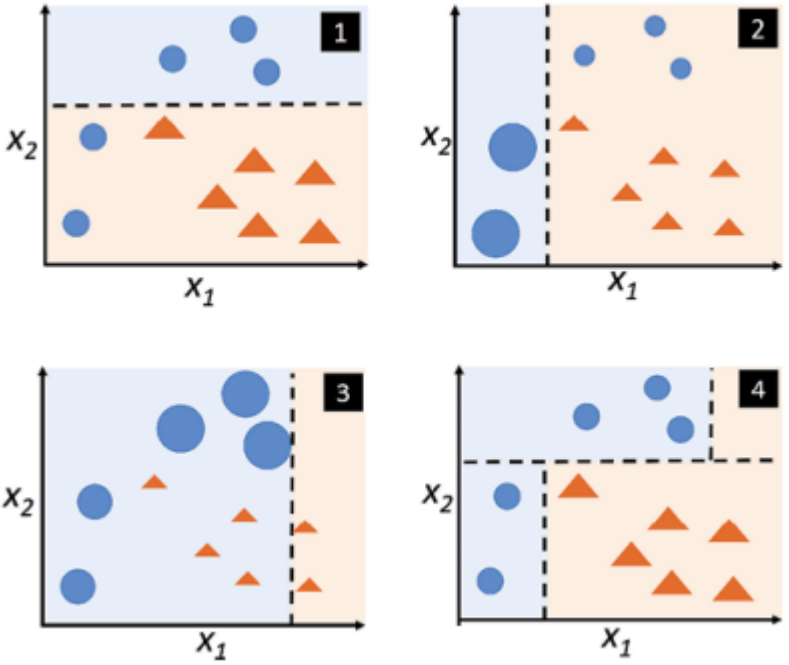


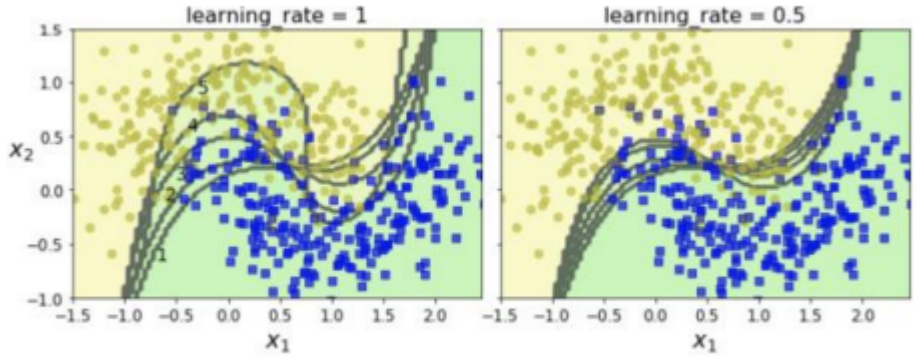
그림7-7 샘플의 가중치를 업데이트하면서 순차적으로 학습하는 에이다부스트 동작

7.5.1(참고) 에이다부스트



7.5.1.1(예제) 에이다부스트

학습률 반으로 낮춤 = 잘못분류된 샘플의 가중치가 반복마다 절반만 높아짐



▲ 그림7-8 연속된 예측기의 결정 경계

7.5.1.2 에이다부스트 알고리즘

(1) 가중치가 적용된 에러율 r_1

$$r_j = \frac{\sum_{i=1}^{m} w^{(i)} \ (where \ \hat{y}_j^{(i)} \neq y^{(i)})}{\sum_{i=1}^{m} w^{(i)}}$$

(2) 예측기의 가중치

$$\alpha_j = \eta \log \frac{1 - r_j}{r_j}$$

(3) 샘플의 가중치 업데이트 – i 번째 샘플의 샘플의 가중치를 업데이트함.

$$w^{(i)} = \begin{cases} w^{(i)} \ (where \ \hat{y}_j^{(i)} = y^{(i)}) \\ w^{(i)} exp(\alpha_j) \ (where \ \hat{y}_j^{(i)} \neq y^{(i)}) \end{cases}$$

(4) 예측 결과 – 가중치 합이 가장 큰 클래스

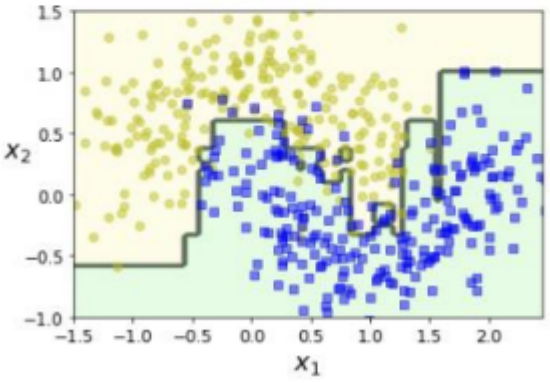
$$\hat{y}(x) = \text{argmax}_k \sum_{j=1}^{N} \alpha_j \ (where \ \hat{y}(x) = k)$$

7.5.1.3 에이다부스트 실습

```python
from sklearn.ensemble import AdaBoostClassifier

ada_clf = AdaBoostClassifier(
    DecisionTreeClassifier(max_depth=1), n_estimators=200,
    algorithm="SAMME.R", learning_rate=0.5, random_state=42)
ada_clf.fit(X_train, y_train)
```

```python
plot_decision_boundary(ada_clf, X, y)
```



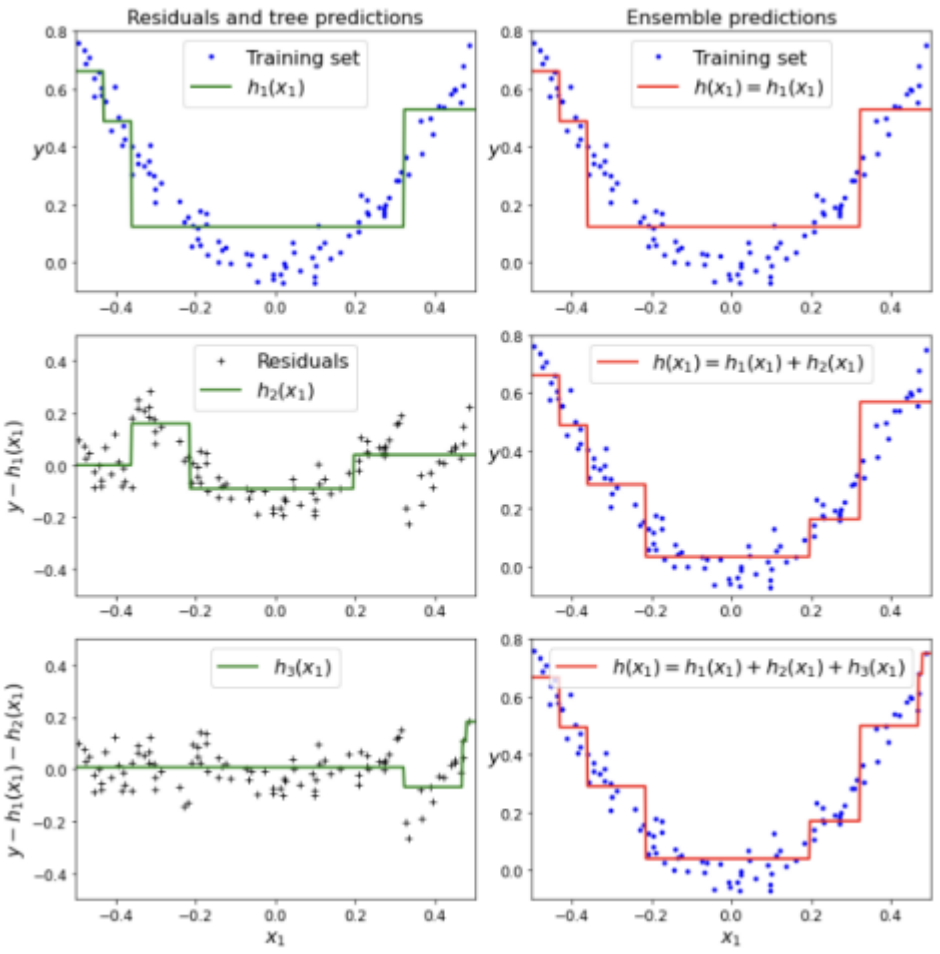
7.5.2 그레이디언트 부스팅

이전 학습기에 의한 오차 보정위해 새로운 예측기를 순차적으로 추가 + 이전 예측기가 만든 잔여 오차(residual error)에 대해 새로운 예측기를 학습

잔여 오차 : 예측값과 실제값 사이의 오차

→ 줄이는 방향으로 모델학습, 경사하강법 사용

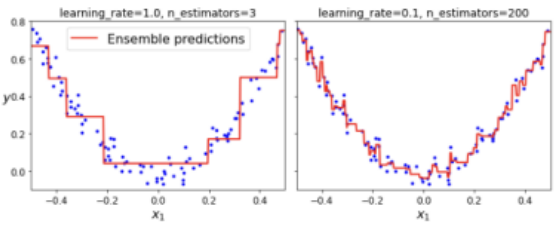
7.5.2.1 그레이디언트 부스팅 적용



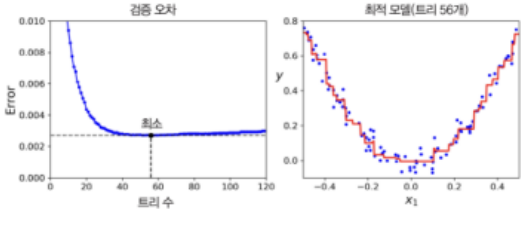
7.5.2.2 GBRT 앙상블 훈련

축소규제: 학습률을 낮게 = 많은 수의 의사결정나무 필요, 성능 좋아짐

최적의 결정트리 수 확인법 : 조기종료 기법



▲ 예측기가 부족한 경우와 너무 많은 경우의 GBRT 앙상블



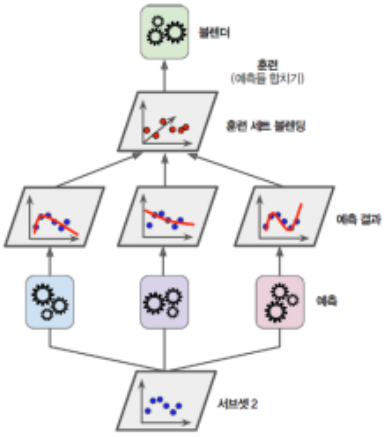
▲ 조기 종료를 사용하여 트리 수 튜닝

7.5.2.3 확률적 그레이디언트 부스팅

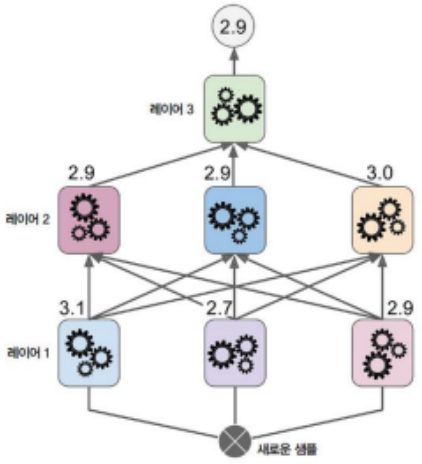
각 결정트리 훈련에 사용할 훈련 샘플의 비율을 지정하여 학습 → 속도 빠름, 편향 증가, 분산 감소

7.6 스태킹

Key Idea : 앙상블에 속한 모든 예측기의 예측을 취합하는 간단한 함수(직접 투표 같은)를 사용하는 대신 취합하는 모델을 훈련시킬 수 없을까?



▲ 그림 7-14 블렌더 훈련



▲ 그림 7-15 멀티 레이어 스태킹 앙상블의 예측