**머신러닝**

**Lab11,12**

**부스팅**

**20135174 정태이**

**●핵심내용 정리**

부스팅 : 여러 개의 약한 학습기를 연결 -> 강한 학습기를 만드는 앙상블 방법

부스팅 방법에는 Adaptive와 Gradient가 있는데

Adaptive 부스팅은 전 모델이 과소적합했던 훈련 샘플의 ‘가중치’를 더 높혀서 학습하는 것이고

Gradient 부스팅은 이전 예측기가 만든 잔여 오차에 새로운 예측기를 학습시킨다.

**●코드:**

from sklearn.ensemble import AdaBoostClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.datasets import make\_moons

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import VotingClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

from sklearn.ensemble import BaggingClassifier

from sklearn.datasets import load\_iris

#데이터세팅

X, y = make\_moons(n\_samples=500, noise=0.30, random\_state=42)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,random\_state=42)

#200개의 아주 얕은 결정 트리를 기반으로 하는 분류기 훈련

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)

y2 = y-tree\_reg1.predict(X)

tree\_reg2 =DecisionTreeRegressor(max\_depth=2,random\_state=42)

tree\_reg2.fit(X,y2)

#세번째 결정트리 모델까지 반복

y3 = y2 - tree\_reg2.predict(X)

tree\_reg3 = DecisionTreeRegressor(max\_depth=2,random\_state=42)

tree\_reg3.fit(X,y3)

#세 개의 트리를 포함하는 앙상블 모델에 새로운 샘플에 대한 예측을 만든다

X\_new = np.array([[0.4]])

y\_pred = sum(tree.predict(X\_new) for tree in (tree\_reg1, tree\_reg2, tree\_reg3))

print(y\_pred)

from sklearn.ensemble import GradientBoostingRegressor

#사이킷런 모듈을 사용하기

gbrt = GradientBoostingRegressor(max\_depth=2, n\_estimators=3,learning\_rate=1.0, random\_state=42)

gbrt.fit(X,y)

y\_pred\_gbrt = gbrt.predict(X\_new)

print(y\_pred\_gbrt)

from sklearn.metrics import mean\_squared\_error

#데이터 세트를 나눠준다

X\_train, X\_val, y\_train, y\_val = train\_test\_split(X,y,random\_state=49)

#Gradient 부스팅 훈련

gbrt = GradientBoostingRegressor(max\_depth=2, n\_estimators=120, random\_state=42)

gbrt.fit(X\_train, y\_train)

#오차검증 측정

errors = [mean\_squared\_error(y\_val, y\_pred)

for y\_pred in gbrt.staged\_predict(X\_val)]

bst\_n\_estimators = np.argmin(errors)

gbrt\_best = GradientBoostingRegressor(max\_depth=2, n\_estimators=bst\_n\_estimators, random\_state=42)

gbrt\_best.fit(X\_train, y\_train)

print(np.argmin(errors))

print(np.min(errors))

gbrt = GradientBoostingRegressor(max\_depth=2, warm\_start=True, random\_state=42)

#다섯 번의 반복동안 검증 오차가 향상되지 않으면 훈련 종료

min\_val\_error = float("inf")

error\_going\_up =0

for n\_estimators in range(1,120):

gbrt.n\_estimators = n\_estimators

gbrt.fit(X\_train, y\_train)

y\_pred = gbrt.predict(X\_val)

val\_error = mean\_squared\_error(y\_val, y\_pred)

if val\_error < min\_val\_error:

min\_val\_error = val\_error

error\_going\_up =0

else:

error\_going\_up +=1

if error\_going\_up == 5:

break

print(gbrt.n\_estimators)

print("최소 검증 MSE:",min\_val\_error)

**결과화면 :**







