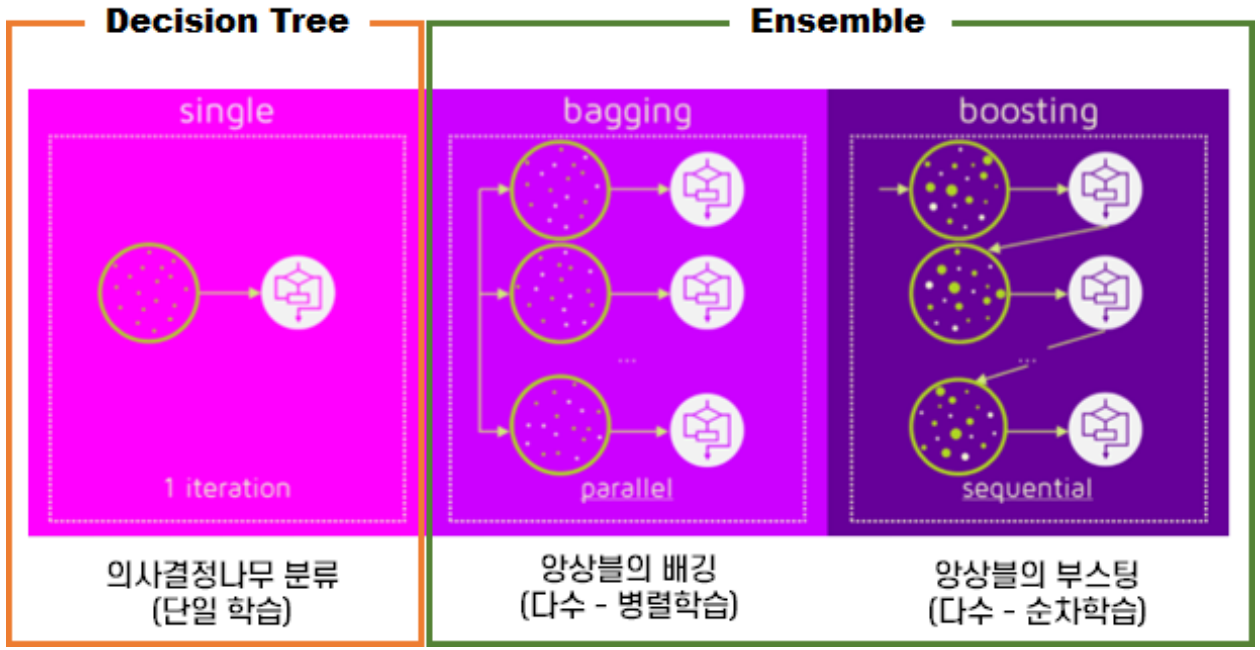
1. 앙상블 개념

여러 가지 우수한 학습 모델을 조합해 예측력을 향상시키는 모델

Assume :

〈성능 비교〉

→ *정확도가 높은 하나의 모델* << **정확도가 낮은 여러 개의 모델의 조합**



단,
1. 모델 결과의 해석이 어려움
2. 긴 예측 시간 소모

출처 : https://joyfuls.tistory.com/61

1. 앙상블 개념

분류	Bagging	Boosting
공통점	전체 data set으로부터 복원 랜덤 샘플링으로 train set 생성	
차이점	병렬 학습	순차 학습
장점	과대 적합에 유리	높은 정확도
단점	특정 영역에서 낮은 정확도	Outlier에 취약
특징	균일한 확률 분포에 의해 train set 생성	분류하기 어려운 train set 생성

→ Random Forest

→ AdaBoost, Gradient Boost

출처 : https://joyfuls.tistory.com/61

3. Bagging

< Bagging 알고리즘 >

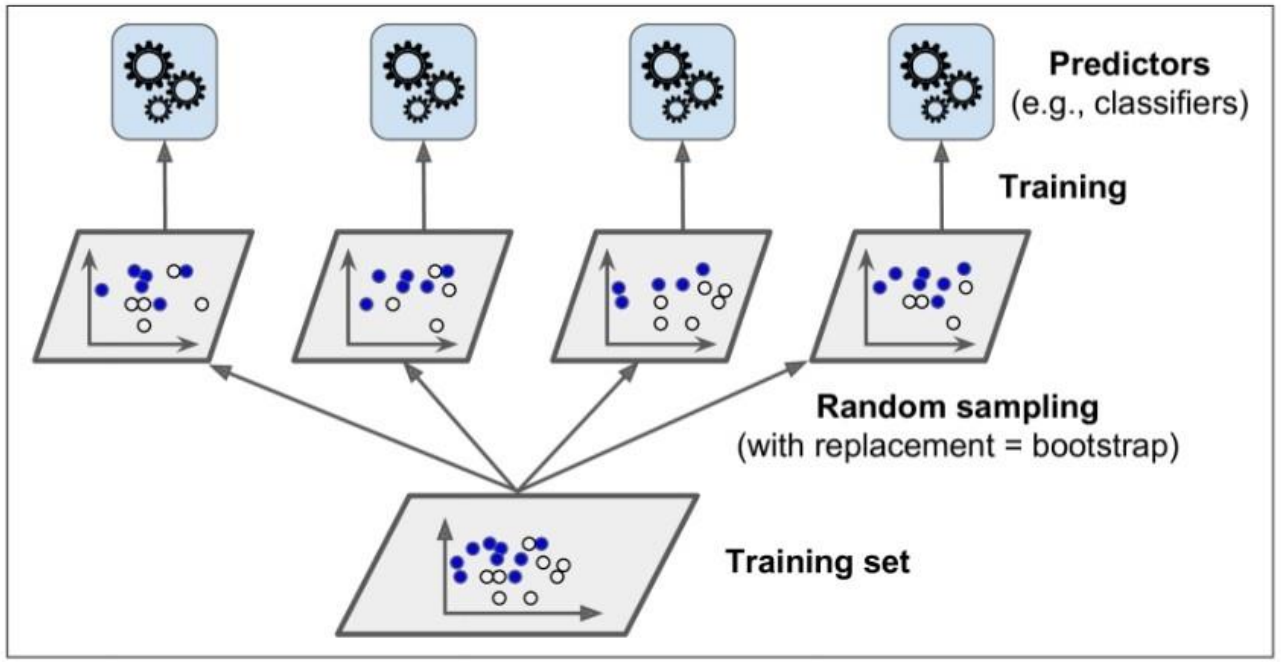
목적 : Variance ↓

Dataset에 **무작위성** 부여 → 더 **견고**하고 **안정성** 있는 모델 (성능 향상)

1. Bootstrap (복원 랜덤 샘플링) : 한 가지 **분류 모델을 여러 개** 만들어서 서로 다른 train data로 학습
2. Aggregating : 동일한 test data에 대해서 **여러 분류 모델이 예측한 값**들을 조합

→ **투표**를 통해 적절한 예측값으로 최종 결론 도출

∴ **과대 적합**이 되기 쉬운 모델을 예측할 때 적합



출처 : https://deepflowest.tistory.com/164

3. Bagging

< Bootstrap > : Random sampling with replacement에 기반한 통계 검정(혹은 추정)

>> 일반적으로 관측된 random sample에서 resampling을 수행하는 case resampling 기법
→ 통계량(추정량)의 분포를 구함

original sample X = {x1, ..., xn} → bootstrap sample Y = {y1, ..., ym}

X의 각 관측치가 선택 될 확률이 같도록
X에서 대체하여 선택한 샘플

→ **장점 : x의 분포에 대한 가정 X**

∴ **주로 모집단의 분포를 구할 수 없는 경우에 분포를 추정하기 위해 사용**

** **Bagging에서의 적용** : 추정 혹은 예측 자체의 성능을 개선시키기 위해 bootstrap 사용
즉, 주어진 train data로부터 bootstrap sample을 resampling한 후, 각 bootstrap sample 내에서 fit된 모형들의 예측 값들의 평균 도출 (classification : 다수결 원칙)

출처 : https://lee-jaejoon.github.io/stat-bootstrap/

3. Bagging

```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier

bag_clf=BaggingClassifier(
    DecisionTreeClassifier(random_state=42), n_estimators=500,
    max_samples=100, bootstrap=True, random_state=42
)

bag_clf.fit(X_train, y_train)
y_pred=bag_clf.predict(X_test)


from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, y_pred))
# 0.904
```

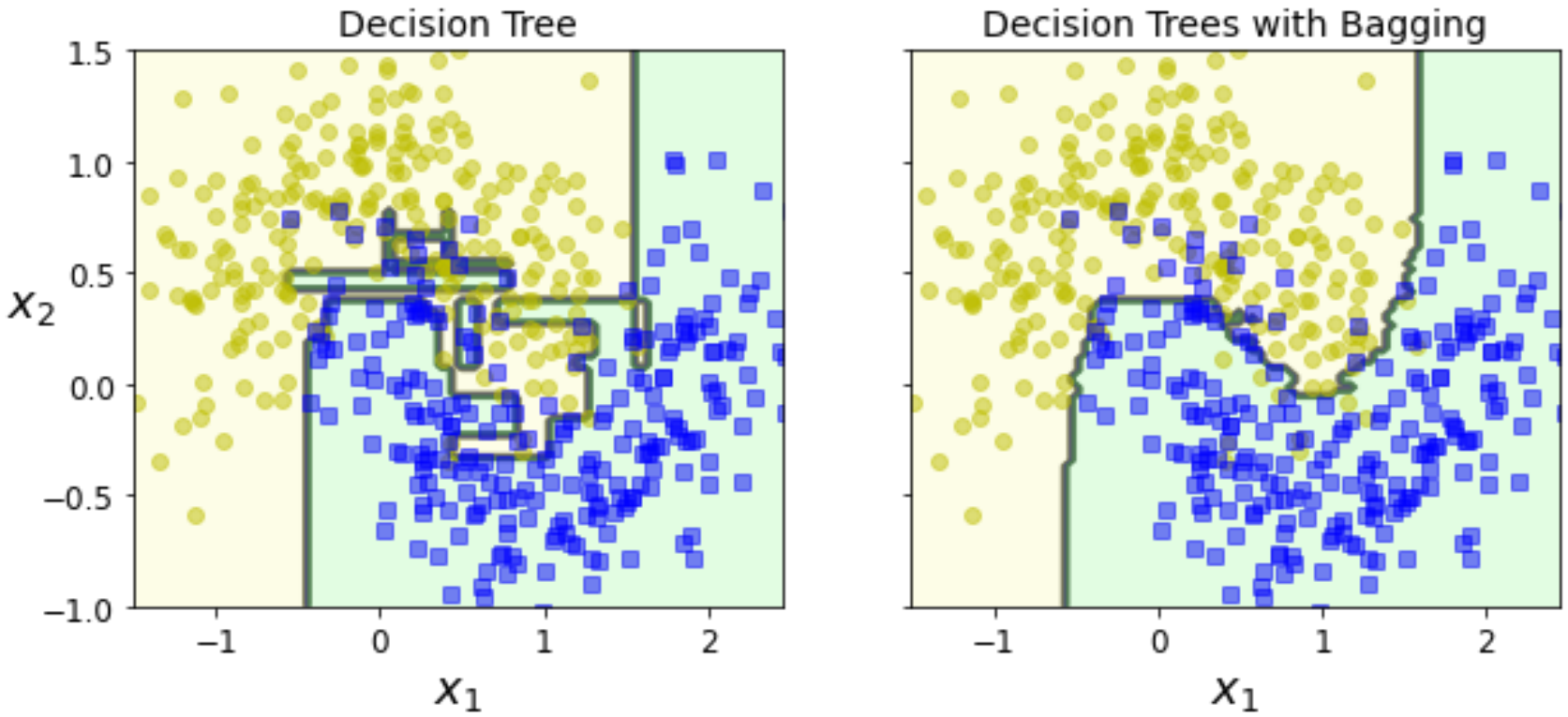
결정 트리를 기반 모델로 지정

```
tree_clf=DecisionTreeClassifier(random_state=42)

tree_clf.fit(X_train, y_train)
y_pred_tree=tree_clf.predict(X_test)
print(accuracy_score(y_test, y_pred_tree))
# 0.856
```

3. Bagging

〈 2차원 평면에서의 결정 경계 〉



3. Bagging

< OOB 평가 : Out Of Bag >

- **OOB sample :** Original sample에서 **훈련에 전혀 사용되지 않는 sample**

original sample X = {x1, ..., xn} → bootstrap sample Y = {y1, ..., ym}에서 (일반적으로 m=n)

n개의 sample 중 bootstrap sample로 선택되지 않을 확률 $= \lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{\boldsymbol{m}} = \lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{\boldsymbol{n}} = e^{-1} \approx 37\%$

- **OOB 평가 :** OOB sample로 앙상블 모델을 평가하는 방식

→ **장점 : 훈련 세트를 효율적으로 사용**

3. Bagging

< OOB 평가 : Out Of Bag >

```
bag_clf=BaggingClassifier(
    DecisionTreeClassifier(random_state=42), n_estimators=500,
    bootstrap=True, oob_score=True, random_state=40
)

bag_clf.fit(X_train, y_train)
print(bag_clf.oob_score_)
# 0.8986666666666666

bag_clf.oob_decision_function_[0]
# [0.32275132, 0.67724868]
```

OOB 평가 실시

```
from sklearn.metrics import accuracy_score

y_pred=bag_clf.predict(X_test)
print(accuracy_score(y_test, y_pred))
# 0.912
```