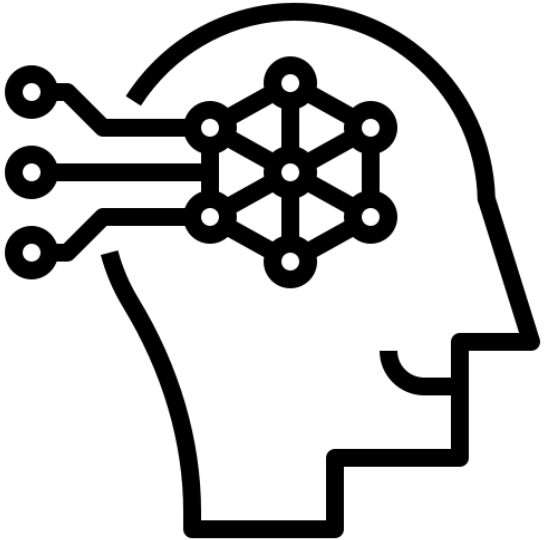
트리,배깅,앙상블

15기 분석 김은선
16기 분석 김연선

Contents

Decision Tree

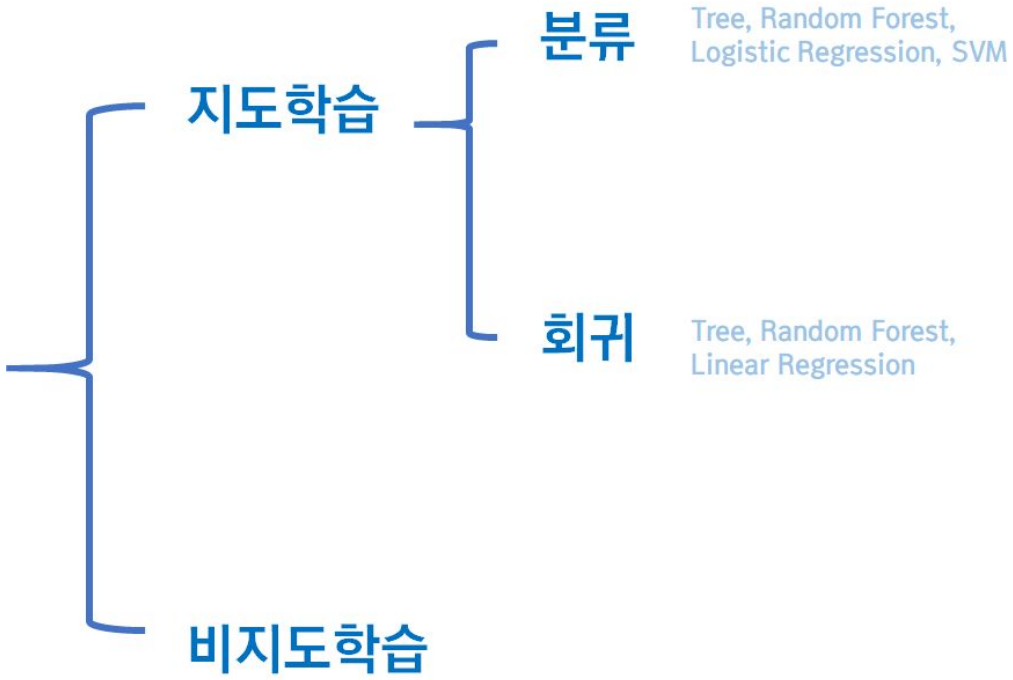
머신러닝 개요



사람의 감독하에 학습이 진행되는가?👨‍🏫

지도, 비지도, 준지도, 강화학습

사람의 감독하에 학습이 운영되는가?



Decision Tree란?

나무구조로 도표화하여 분류 및 예측을 수행하는 머신러닝 알고리즘🌴

Decision Tree란?

✔의사결정나무 특징

1. 분류, 회귀 모두 가능한 머신러닝 알고리즘

분류 - DecisionTreeClassifier
회귀 - DecisionTreeRegressor

2. 질문을 던져 정답/오답에 따라 대상을 좁혀나감

Decision Tree 모델 개요

Decision Tree 모델 개요

input data & output data

Decision Tree 모델 개요

input data & output data　　데이터를 2개 이상으로 분할

데이터가 균일해지도록 분할

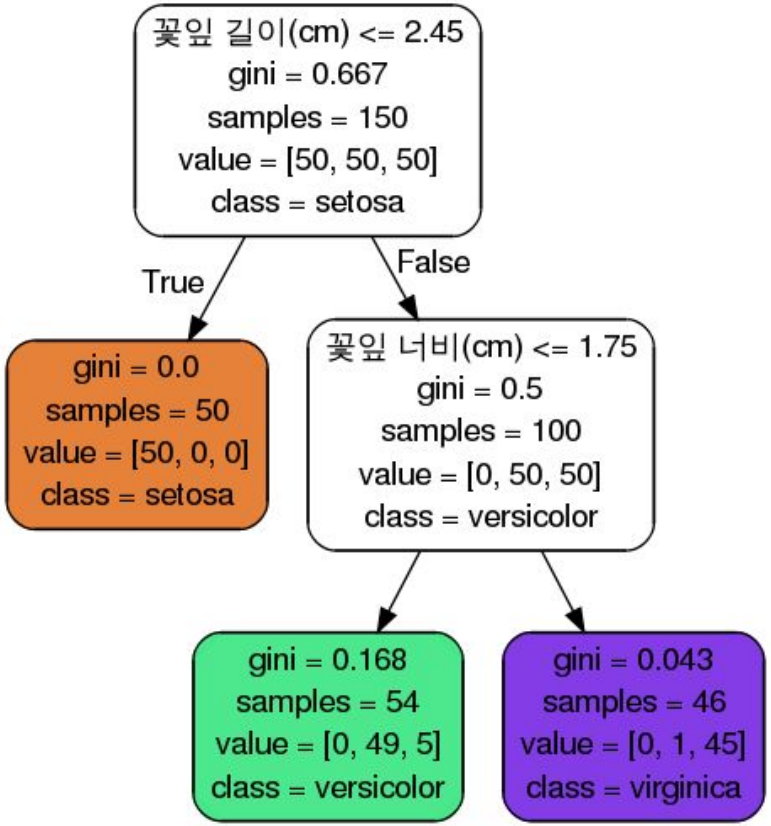
- **분류** : 비슷한 범주를 가지고 있는 관측치끼리 균일해지도록
- **회귀** : 비슷한 수치를 가지고 있는 관측치끼리 균일해지도록

Decision Tree 모델 개요

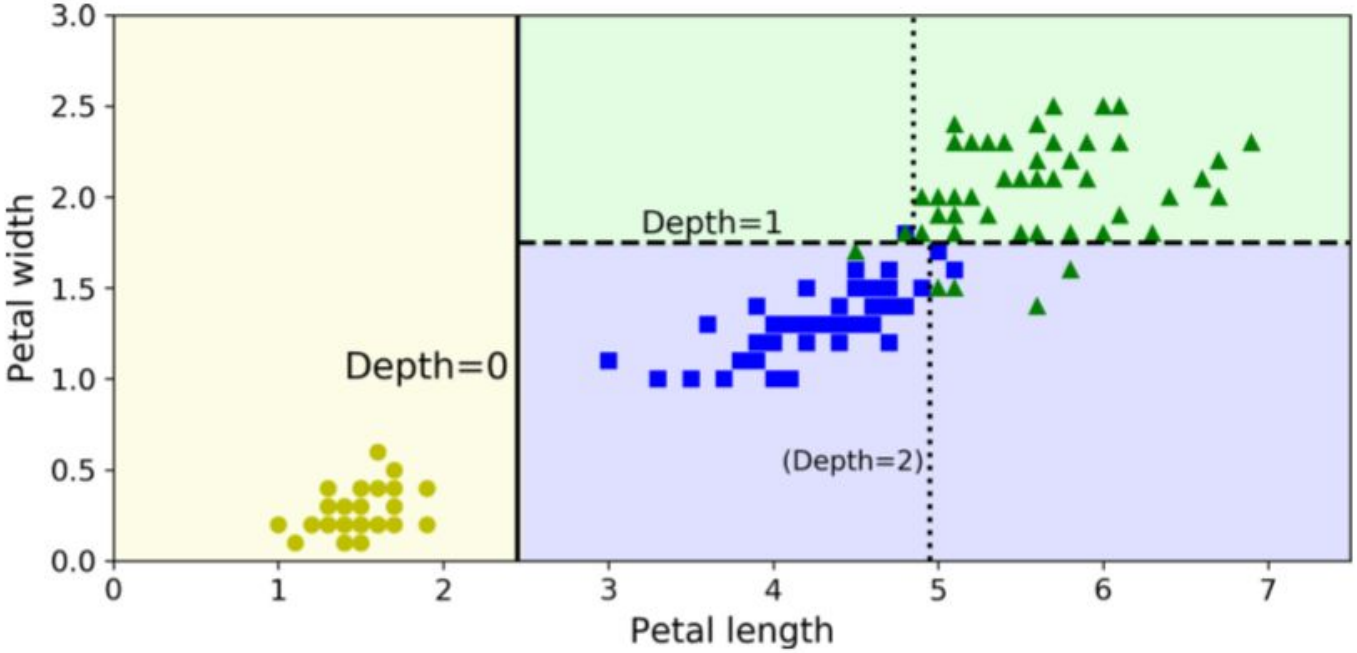
input data & output data

데이터를 2개 이상으로 분할

Decision Tree 모델 개요



Decision Tree 모델 개요



Decision Tree 형성 과정

적절한 '분할규칙'과 '정지규칙'을 지정하고 '예측값'을 할당

분리규칙

정지규칙

가지치기

예측값
할당

Decision Tree 형성 과정

분리규칙

정지규칙

가지치기

예측값 할당

- 부모마디에서 자식 마디를 생성하는 기준
- 순도 / 불순도에 의해 목표 변수를 구별

Decision Tree 형성 과정

CART 훈련 알고리즘

Classification and Regression Tree

[기본원리]

훈련 세트를 하나의 특성 k의 임곗값 t_k를 사용해 서브셋으로 나누기

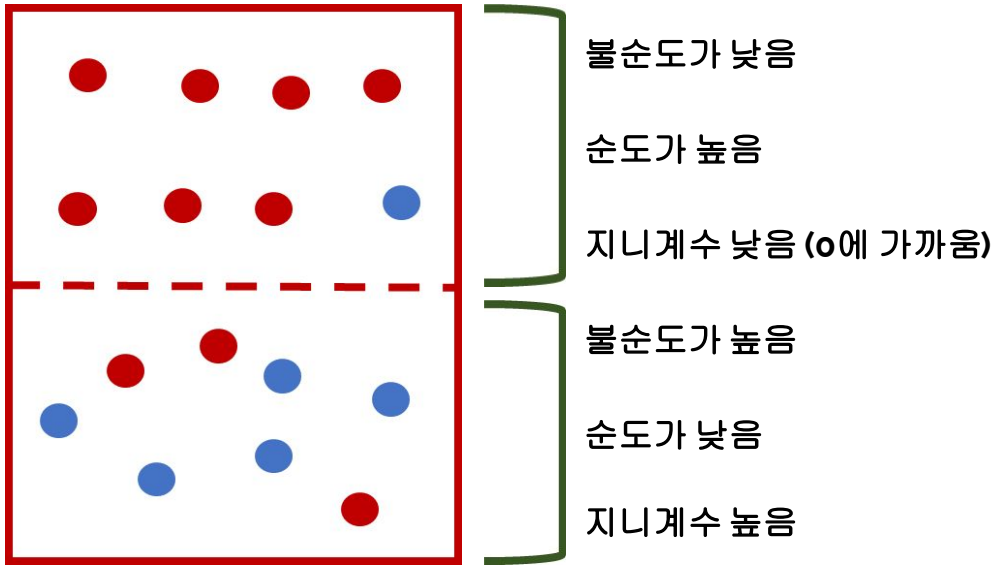
[분류에 따른 CART 비용함수]

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

m = 전체 샘플 수 (m_{left} : 왼쪽 서브셋일 불순도, m_{right} : 오른쪽 서브셋의 불순도)
G : 불순도 (G_{left} : 왼쪽 샘플 수 , G_{right} :오른 쪽 샘플 수)

Decision Tree 형성 과정

순도 / 불순도 / 지니계수



✔ 불순도를 최소화 하는 방향으로 학습 진행

Decision Tree 형성 과정

엔트로피

[정보이론]

- 데이터를 정량화하기 위한 응용수학 분야 중 하나
- 정보량이 높다 = 어떤 일이 일어날 확률이 낮다. 불확실하다.
- 이때, 정보량이 높은 문장이 맞을수록 해당 정보의 중요도는 높아진다.

[엔트로피]

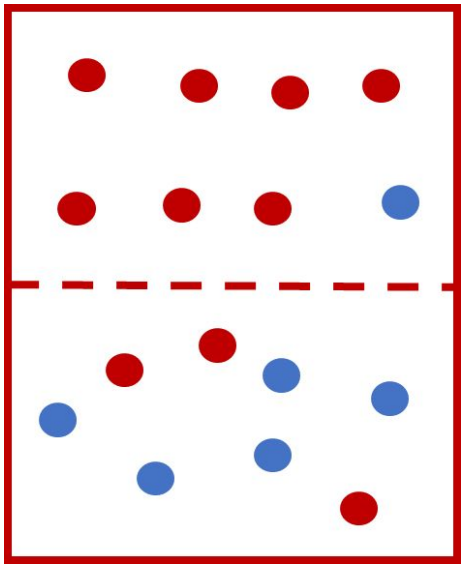
: 정보량의 평균

: 분자의 무질서함을 측정하는 개념

$$Entropy(A) = -\sum_{k=1}^{m} p_k \log_2 (p_k)$$

Decision Tree 형성 과정

엔트로피



$$Entropy(A) = -\sum_{k=1}^{m} p_k \log_2 (p_k)$$

분할 전

$$Entropy(A) = -\frac{10}{16}\log_2 (\frac{10}{16}) - \frac{6}{16}\log_2 (\frac{6}{16}) \approx 0.95$$

분할 후

$$Entropy(A) = \sum_{i=1}^{d} R_i \left(-\sum_{k=1}^{m} p_k \log_2 (p_k) \right)$$

$$Entropy(A) = 0.5 \times \left(-\frac{7}{8}\log_2 (\frac{7}{8}) - \frac{1}{8}\log_2 (\frac{1}{8}) \right) + 0.5 \times \left(-\frac{3}{8}\log_2 (\frac{3}{8}) - \frac{5}{8}\log_2 (\frac{5}{8}) \right) \approx 0.75$$

Decision Tree 형성 과정

분리규칙 | **정지규칙** | 가지치기 | 예측값 할당

[더 이상 분리가 일어나지 않는 기준]

1) 더이상 분리해도 불순도가 줄지 않을 때
2) 자식마디에 남은 sample 수가 너무 적을 때
3) 분석자가 지정한 규제 매개변수에 도달했을 때

Decision Tree 형성 과정

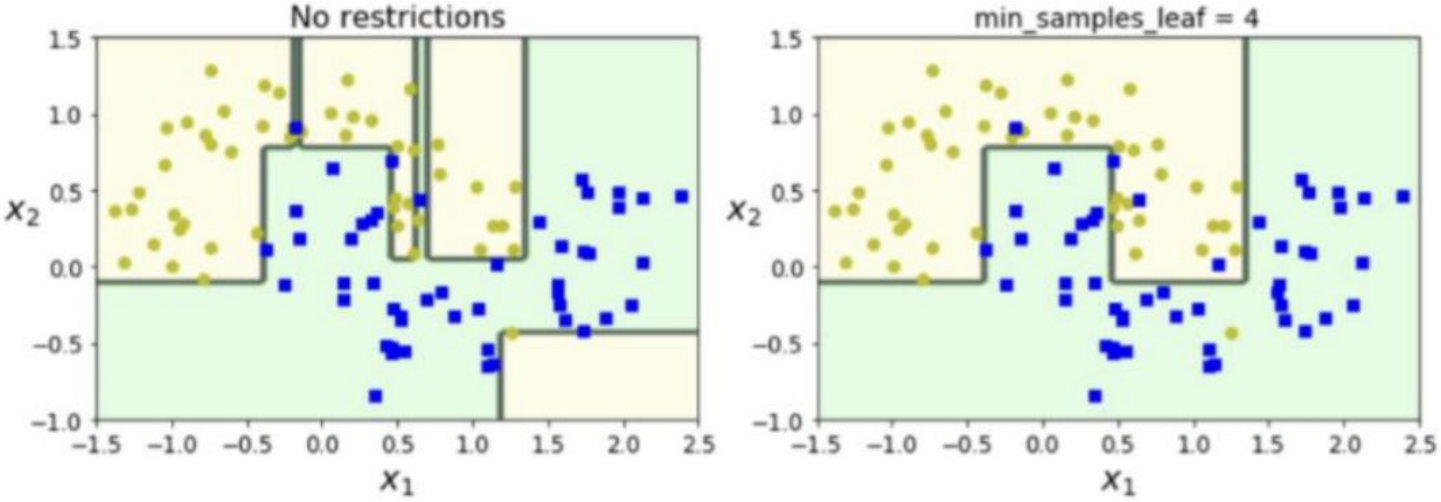
규제 매개변수

결정 트리는 대체로 데이터의 대한 제약 사항이 매우 적지만,
오버피팅의 위험성이 커 규제 매개변수를 활용해 제약을 걸어주는 것이 바람직함

max_depth: 최대 기피 설정
min_samples_split: 분할되기 위해 노드가 가져야하는 최소 샘플 수
min_samples_leaf: 리프 노드가 가지고 있어야 하는 최소 샘플 수
max_leaf_nodes: 리프 노드의 최대 수
max_featrues: 각 노드에서 분할에 사용할 특성의 최대 수

Decision Tree 형성 과정

규제 매개변수



왼쪽은 규제가 없는 상태, 오른 쪽은 리프노드의 최소 샘플 수를 4개로 제한한 상태

=> 오른쪽은 과적합을 피함

Decision Tree 형성 과정

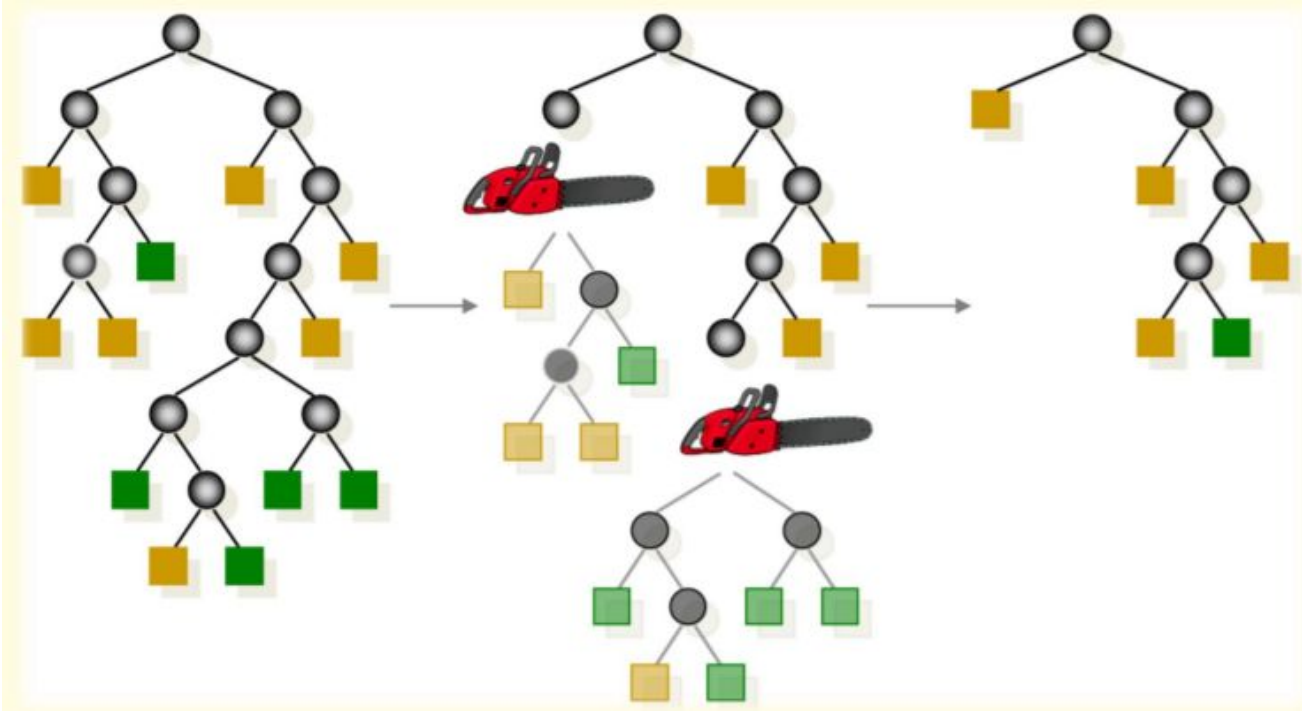
분리규칙 → 정지규칙 → **가지치기** → 예측값 할당

[부적절한 마디를 잘라내 모양 단순화]

:depth가 깊어질 수록 오버피팅 위험성 높음
:불필요한(부적절한) 마디 제거하는 과정
:데이터를 버리는 것 X 합치는 과정 O (merge)

Decision Tree 형성 과정

가지치기



Decision Tree 형성 과정

분리규칙 → 정지규칙 → 가지치기 → 예측값 할당

- 예측값 할당
(분류: class 예측 vs 회귀: 특정 값 예측)
- 타당성 평가 (cross validation 등을 통해 트리 모델 평가)
- 해석 및 예측 (생성한 tree에 새로운 데이터 대입 => 확인)

Decision Tree 장단점

✔ 직관적
✔ 이상치, 노이즈에 큰 영향받지 않음
✔ 높은 모델 해석력
✔ 연속형, 범주형 데이터 모두 처리 가능
✔ 균일도에만 초점 가능
(스케일링, 정규화 불필요)

✔ 일반화가 어려움 (학습데이터에 따른 차이 큼)
✔ 모델 variance 가 높음
✔ 오버피팅 가능성 매우 높음

=> **Random Forest의 등장 계기**
(트리들에서 만든 예측을 평균, 불안전성 극복!)

https://medium.com/ml-research-lab/ensemble-learning-the-heart-of-machine-learning-b4f59a5f97

Iris 데이터로
Decision Tree 실습

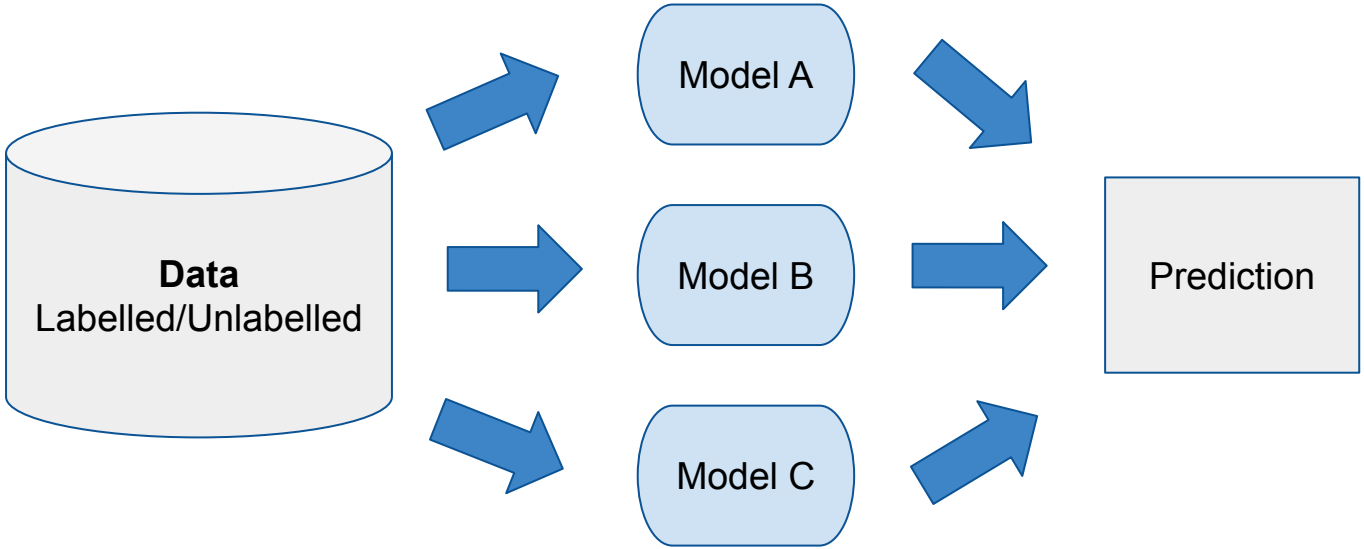
Ensemble

Ensemble

https://www.liveabout.com/ensembles-245670

Ensemble

❑ 앙상블이란?: 여러 모델의 결과를 바탕으로 새로운 모델을 만들어, 더 높은 예측력을 보여주는 방법



https://medium.com/ml-research-lab/ensemble-learning-the-heart-of-machine-learning-b4f59a5f97

Ensemble

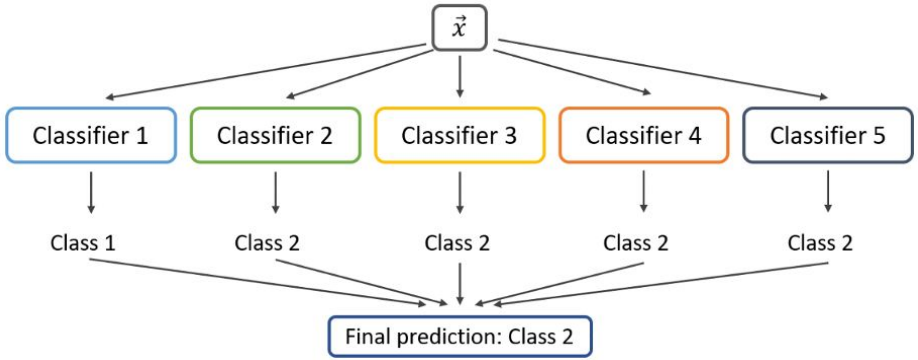
❑ **앙상블의 종류:**

1. 보팅 (Voting): 서로 다른 알고리즘의 결과물에 대해 투표로 결정.
   a. Hard Voting
   b. Soft Voting
2. 배깅 (Bagging): 복원 추출하여 각 모델을 학습시켜 결과를 집계.
   a. Random Forest
3. 부스팅 (Boosting): 올바르게 예측되지 못한 데이터에 가중치를 두어 재학습.
   a. AdaBoost
   b. CatBoost
   c. GBM
   d. XGBoost
4. 스태킹 (Stacking): 서로 다른 알고리즘들을 조합해서 장점을 강화하고 약점을 보완.

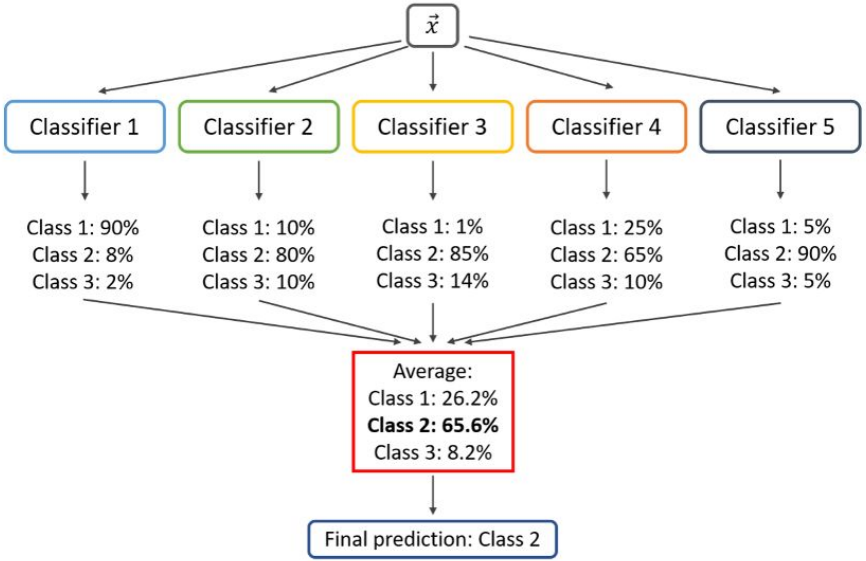
Voting

- 보팅이란?: 서로 다른 알고리즘의 결과물에 대해 투표로 결정
  - 방법: Hard voting, Soft voting

<Hard Voting> - 다수결



<Soft Voting> - 클래스별 확률 평균 계산



https://www.kaggle.com/fengdanye/machine-learning-6-basic-ensemble-learni

Voting

sklearn.ensemble.VotingClassifier

class sklearn.ensemble.**VotingClassifier**(*estimators*, *, *voting='hard'*, *weights=None*, *n_jobs=None*, *flatten_transform=True*, *verbose=False*) [source]

sklearn.ensemble.VotingRegressor

class sklearn.ensemble.**VotingRegressor**(*estimators*, *, *weights=None*, *n_jobs=None*, *verbose=False*) [source]

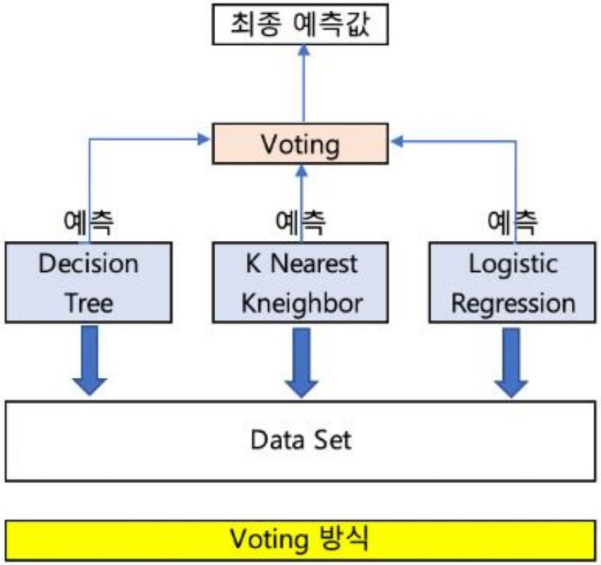
```
eclf_soft = VotingClassifier(estimators=[
    ('lr', clf1), ('rf', clf2), ('gnb', clf3)], voting='soft')
eclf_soft = eclf_soft.fit(X, y)

eclf_hard = VotingClassifier(estimators=[
    ('lr', clf1), ('rf', clf2), ('gnb', clf3)], voting='hard')
eclf_hard = eclf_hard.fit(X, y)

print('Soft voting 방법:', eclf_soft.predict(X))
print('Hard voting 방법:', eclf_soft.predict(X))
```

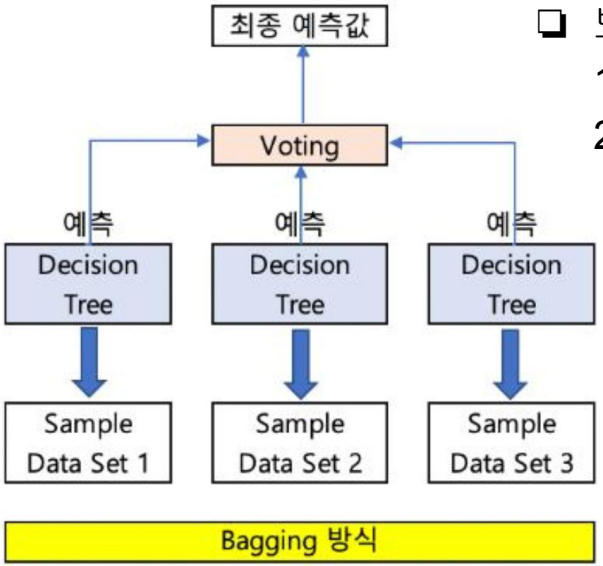
```
Soft voting 방법: [1 1 2 2 2]
Hard voting 방법: [1 1 2 2 2]
```

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html

Bagging

❑ 배깅이란?: 데이터를 복원 추출하여 각 모델을 학습시켜 결과를 집계.





❑ 보팅과의 차이점:

1) 데이터 셋을 샘플링
2) 각 모델은 같은 알고리즘

https://continuous-development.tistory.com/m/179

Bagging

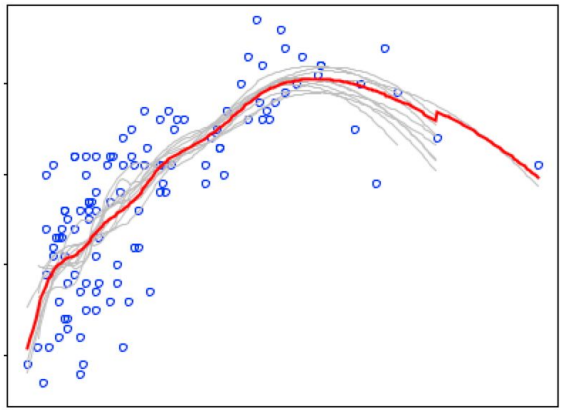
❑ 배깅의 목적:

1) 데이터가 고르지 않을 때, 예측 모델의 변동성 감소.
2) 오버피팅을 방지.

클래스 A: 20
클래스 B: 100

⇨ 불균형 데이터 → 가중치 주기
→ 데이터 샘플링 → Bootstrapping

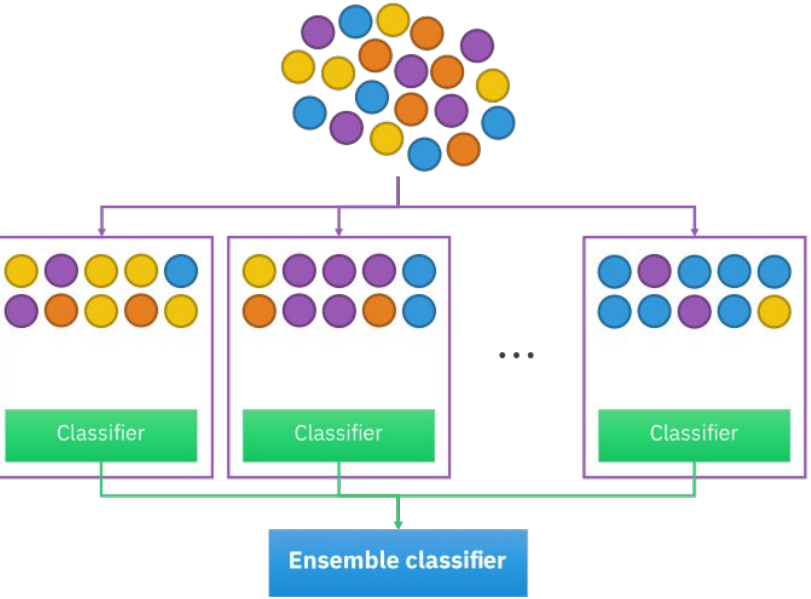
* Boostrap: 통계학에서 사용,
중복을 허용하여 random sampling을 적용하는 기법.



⇨ 오버피팅 (high variance, low bias) 방지

https://www.quora.com/How-does-bagging-avoid-overfitting-in-Random-Forest-classification

Bagging



Original Data

Bootstrapping

Aggregating

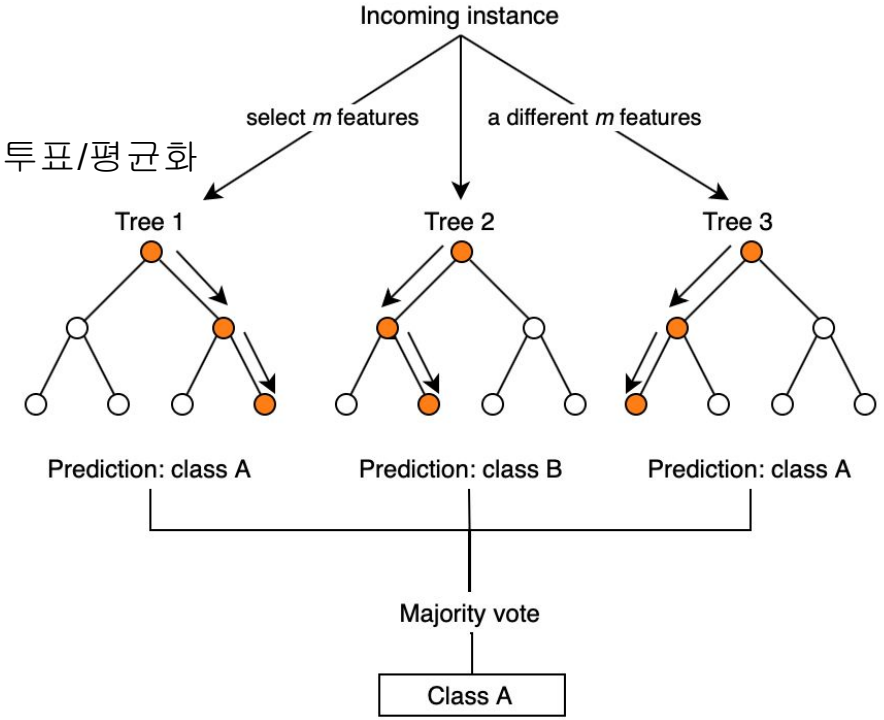
Bagging

- 배깅 원리:
  1) 데이터에서 중복을 허용하게 랜덤 샘플링
  2) 각 샘플링된 데이터를 바탕으로 모델 학습
  3) 각 모델을 집계하여 최종 결과 도출
- categorical data → 투표 방식
- continous data → 평균 집계

Random Forest

- 결정트리 기반 알고리즘.
- 과적합 확률이 큰 여러 개의 결정트리의 결과를 투표/평균화
  → 과접한 양 줄이기.

- 장점: 1. 빠른 학습 시간
  2. 과적합 방지
  3. 결측치 비율이 높아도 높은 정확도
  4. 피처 중요도 확인 가능
- 단점: 많은 메모리 사용량, 튜닝할 변수 많음.



https://deepai.org/machine-learning-glossary-and-terms/random-forest

Ensemble 실습

앙상블의 부스팅과 스태깅은 다음 세션에!

QnA

감사합니다.