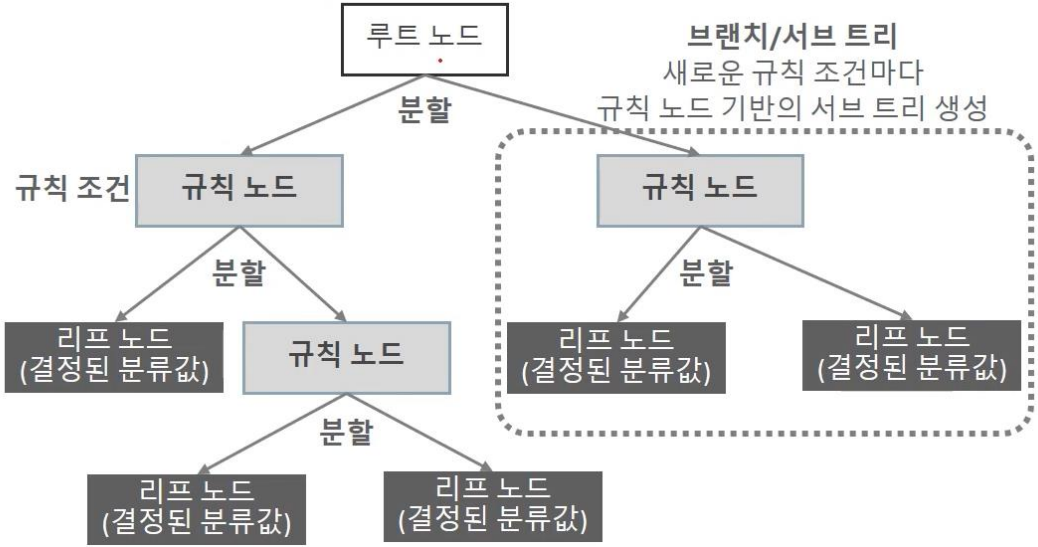
PART02
머신러닝 기본(scikit learning)

RAPA 한국전파진흥협회

결정트리 설명

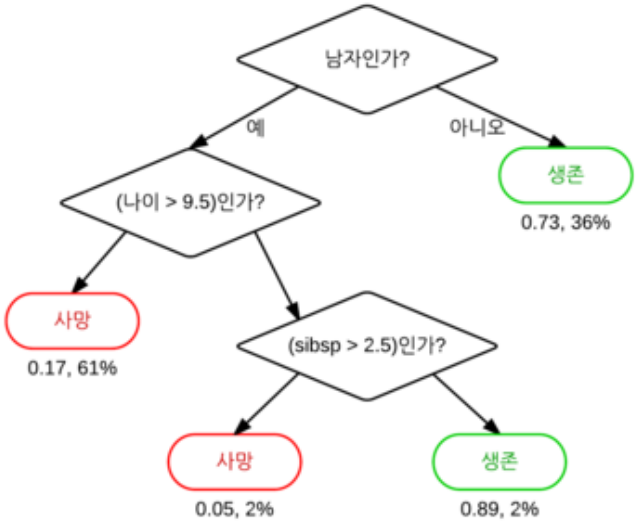
결정트리

- 데이터에 있는 규칙을 학습을 통해 자동으로 찾아내 트리 기반으로 분류 규칙을 만듬(if else 기반 규칙)
- 데이터의 어떤 기준을 바탕으로 규칙을 만들어야 가장 효율적인 분류가 될 것 인가가 알고리즘의 성능을 크게 좌우함.



결정트리(Decision Tree)

- 데이터 마이닝에서 일반적으로 사용되는 방법론
- 몇몇 입력변수를 바탕으로 목표 변수의 값을 예측하는 모델을 생성하는 것을 목표로 함
- 트리 구조로 가지, 자식(child)노드, 잎(leaf) 노드로 구성됨



https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=moonsoo5522&logNo=220888886396

Scikit-learn의 DecisionTree Estimator

- sklearn.tree.DecisionTreeClassifier(분류문제에 사용)
- sklearn.tree.DecisionTreeRegressor(회귀문제에 사용)

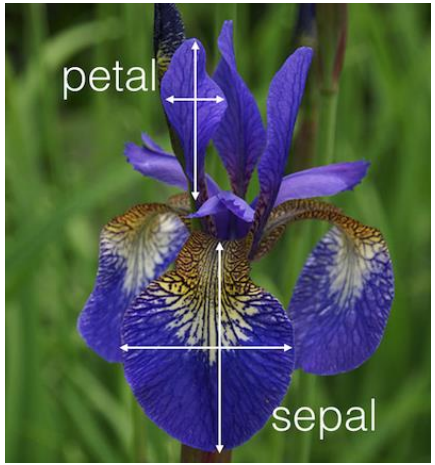
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html#sklearn.tree.DecisionTreeRegressor

붓꽃 데이터 분류 예측 – 지도학습

- 붓꽃 데이터 세트로 붓꽃의 품종 분류 예측하기
- 붓꽃 데이터 세트 : 꽃잎의 길이와 너비, 꽃받침의 길이와 너비

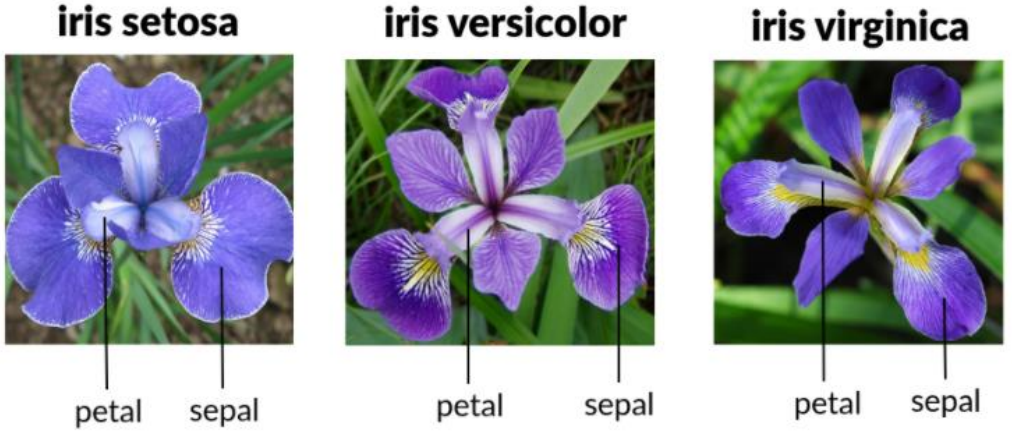
붓꽃 데이터 feature



Petal length
Petal width

Sepal length
Sepal width

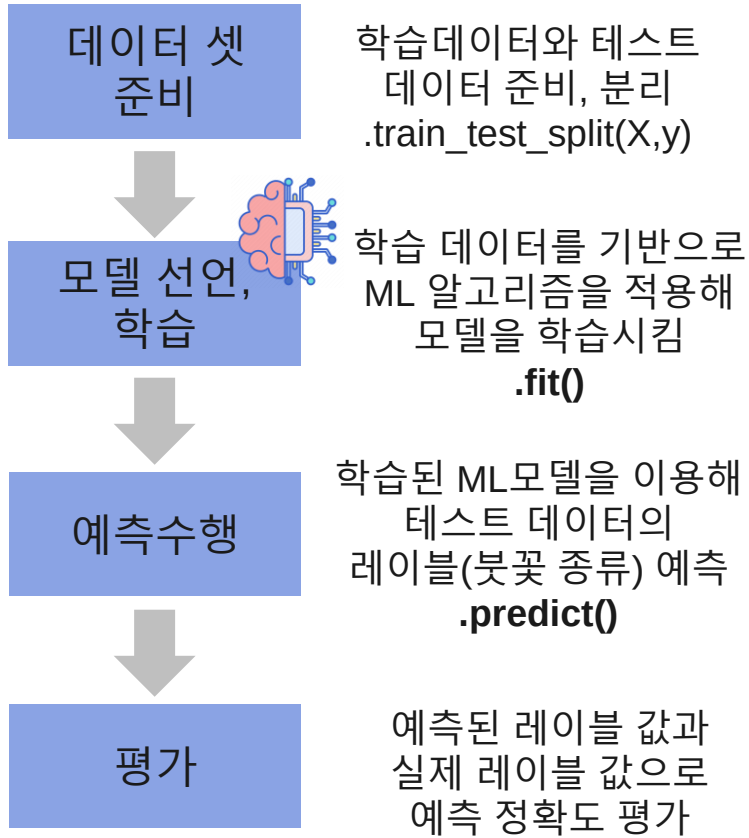
붓꽃 데이터 품종(레이블)



붓꽃 품종 예측하기 프로세스

	피처				레이블
학습 데이터 셋	꽃받침 길이	꽃받침 너비	꽃잎 길이	꽃잎 너비	Iris 꽃 종류
	5.1	3.5	1.4	0.2	Setosa
	2.9	3.0	1.4	0.2	Setosa
	…	…	…	…	…
	6.4	3.5	4.5	1.2	Versicolor
	…	…	…	…	…

	꽃받침 길이	꽃받침 너비	꽃잎 길이	꽃잎 너비	Iris 꽃 종류
테스트 데이터 셋	5.3	3.2	1.1	0.1	?
	4.2	2.0	2.4	0.4	?
	6.5	3.8	5.5	1.1	?
	…	…	…	…	…



학습데이터와 테스트 데이터 준비, 분리
.train_test_split(X,y)

학습 데이터를 기반으로 ML 알고리즘을 적용해 모델을 학습시킴
.fit()

학습된 ML모델을 이용해 테스트 데이터의 레이블(붓꽃 종류) 예측
.predict()

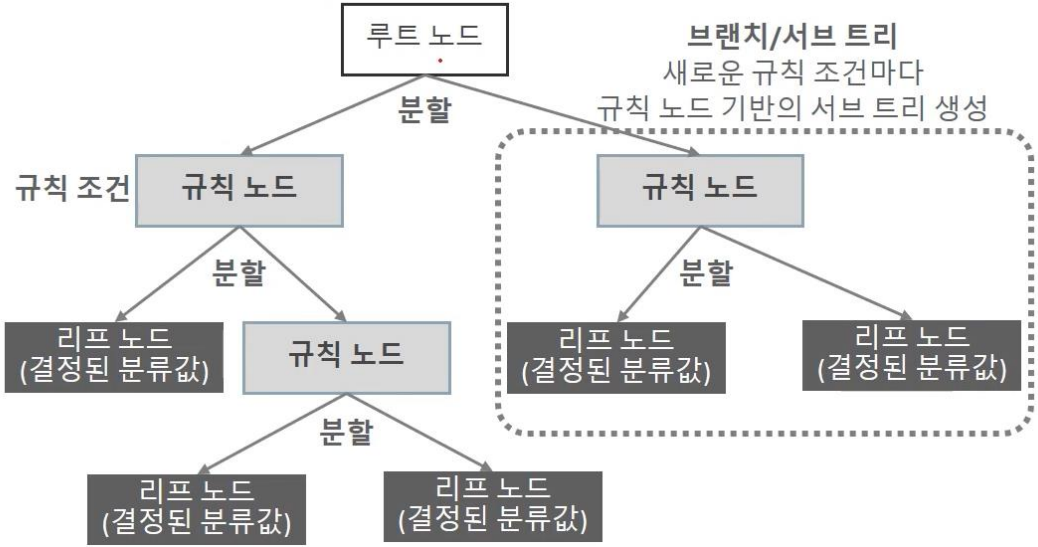
예측된 레이블 값과 실제 레이블 값으로 예측 정확도 평가

결정트리와 앙상블

- 결정 트리
  - 장점
    - 매우 쉽고 유연하게 적용될 수 있는 알고리즘
    - 데이터 스케일링이나 정규화 등의 사전 가공의 영향이 매우 적음
  - 단점
    - 예측성능을 향상시키기 위해 복잡한 규칙 구조를 가져야함.
    - 이로인해 과적합(overfitting)이 발생해 예측성능이 저하 될 수 있음.
- 앙상블 기법
  - 결정트리의 단점이 앙상블 기법에서 장점으로 작용
  - 앙상블 기법은 매우 많은 예측 성능이 상대적으로 떨어지는 학습 알고리즘을 결합해 확률적 보완과 오류가 발생한 부분에 대한 가중치를 계속 업데이트 하면서 예측성능을 향상시킴
  - 결정 트리의 단점을 보완하여 GBM, XGBoost, LightGBM 등으로 발전함

결정트리

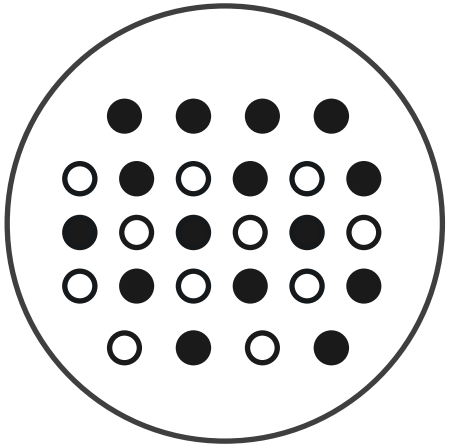
- 데이터에 있는 규칙을 학습을 통해 자동으로 찾아내 트리 기반으로 분류 규칙을 만듬(if else 기반 규칙)
- 데이터의 어떤 기준을 바탕으로 규칙을 만들어야 가장 효율적인 분류가 될 것 인가가 알고리즘의 성능을 크게 좌우함.



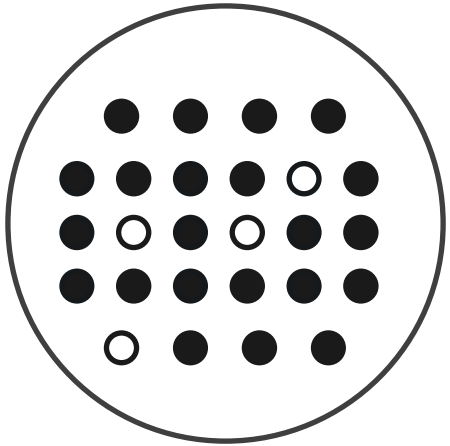
트리 분할을 위한 데이터의 균일도

- 다음 중 가장 균일한 데이터 셋은?

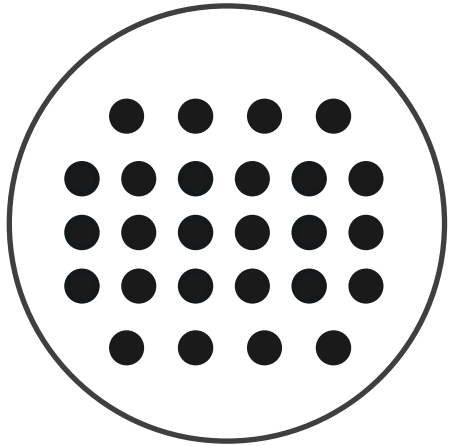
데이터 셋A



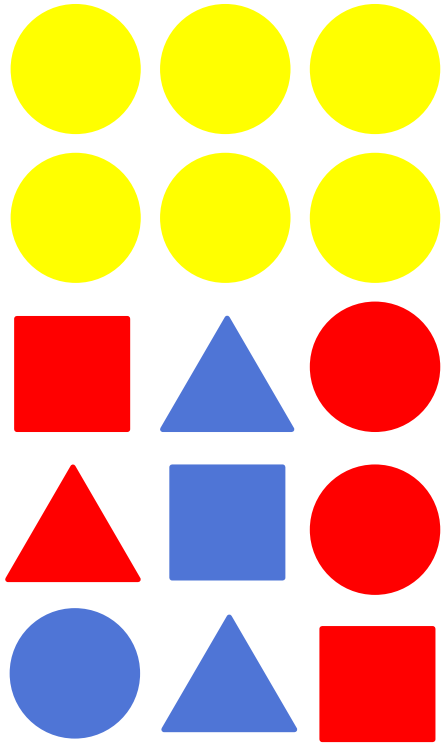
데이터 셋B



데이터 셋C



균일도 기반 규칙 조건



- 노랑색 블록
  - 모두 동그라미로 구성
- 빨강과 파랑 블록
  - 동그라미, 네모, 세모가 골고루 섞여 있음
- 각 레고 블록을 분류하고자 한다면,
  첫번째로 만들어져야 하는 규칙 조건은?

if 색깔 == '노란색'

정보 균일도 측정 방법

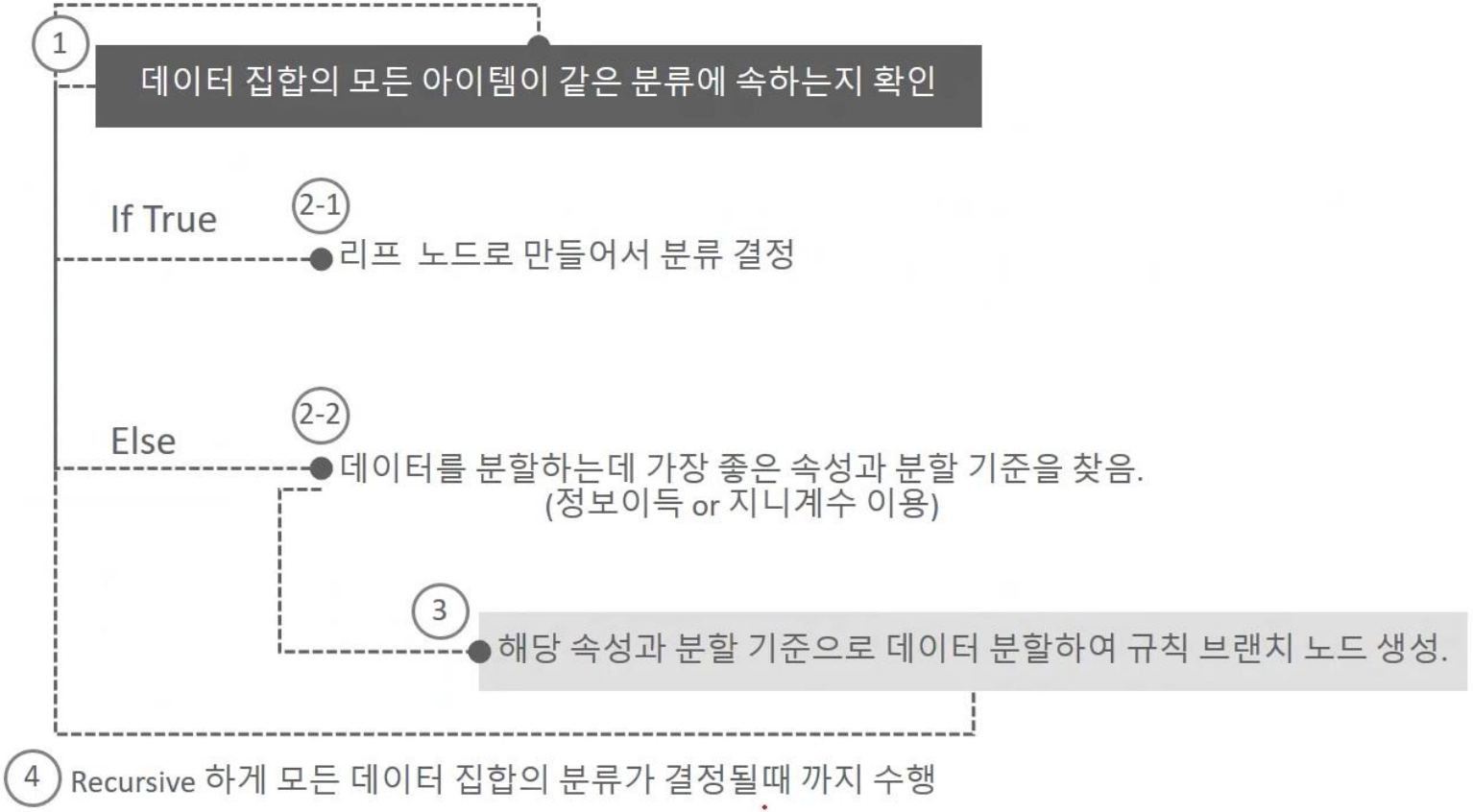
정보 이득(Information Gain) = 1 – 엔트로피 지수

- 엔트로피 개념을 기반으로 함.
- 엔트로피는 주어진 데이터 집합의 혼잡도를 의미함.
  - 서로 다른 값이 섞여 있으면 엔트로피가 높고, 같은 값이 섞여 있으면 낮음.
- 결정 트리는 정보 이득 지수로 분할 기준 정하며, 정보이득이 높은 속성으로 분할 결정 함

지니 계수

- **지니계수는 0이 가장 평등하고, 1로 갈 수록 불평등함.**
- **지니계수가 낮을 수록 데이터 균일도가 높은 것으로 해석됨**
- **계수가 낮은 속성을 기준으로 분할 함.**

결정트리의 규칙 노드 생성 프로세스



결정 트리의 특징

결정 트리 장점	결정 트리 단점
• 쉽다. 직관적이다. • 피처의 스케일링이나 정규화 등의 사전 가공 영향도가 크지 않음.	• 과적합으로 알고리즘 성능이 떨어질 가능성. 이를 극복하기 위해 트리의 **크기를 사전에 제한하는 튜닝 필요**

결정 트리 하이퍼 파라미터

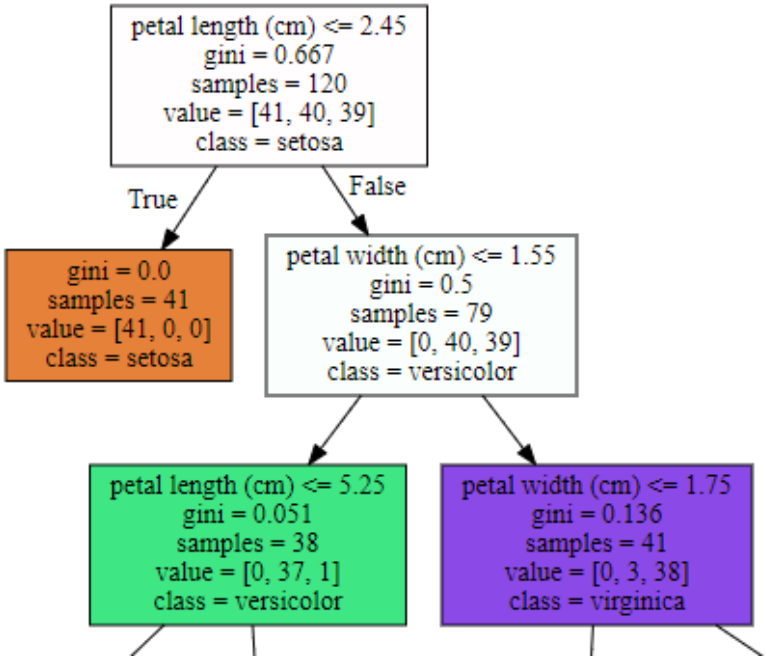
하이퍼 파라미터	설명
max_depth	• **트리의 최대 깊이를 규정** • 디폴트는 None. 완벽하게 클래스 결정 값이 될 때까지 깊이를 계속 키우며 분할하거나 노드가 가지는 데이터 개수가 **min_samples_split 보다 작아질 때까지 계속 깊이를 증가**시킴.
max_features	• **최적의 분할을 위해 고려할 최대 feature 개수** • Default는 None으로 데이터 세트의 모든 feature를 사용해 분할 수행. • Int형 : 대상 feature의 개수, float형:전체 feature 중 대상 feature의 퍼센트임. • 'sqrt' : 전체 feature중 sqrt(전체 피처) 개수 만큼 선정, 'auto' 동일
min_samples_split	• **노드를 분할 하기 위한 최소한의 샘플 데이터 수로 과적합을 제어하는 데 사용됨** • 디폴트는 2이고, 작게 설정할 수록 분할되는 노드가 많아져서 과적합 가능성 증가함, 1로 설정할 경우 분할되는 노드가 많아져서 과적합성 증가
min_samples_leaf	• **말단 노드(Leaf)가 되기 위한 최소한의 샘플 데이터 수** • min_samples_split와 유사하게 과적합 제어 용도. 그러나 비대칭적 데이터의 경우 특정 클래스의 데이터가 극도로 작을 수 있으므로 이 경우는 작게 설정 필요
max_leaf_nodes	• 말단 노드(Leaf)의 최대 개수

결정트리 규칙 생성 시각화 패키지

- Graphviz 패키지 다운로드 설치(c++로 개발됨)
- Graphviz의 파이썬 래퍼 모듈 설치
  - Linux : sudo apt-get install graphviz
  - dot --help : 출력되면 성공
  - scikit learning api 제공
  - from sklearn.tree import export_graphviz

참고 : https://scikit-learn.org/stable/modules/tree.html?highlight=dt

Graphviz의 시각화 노드

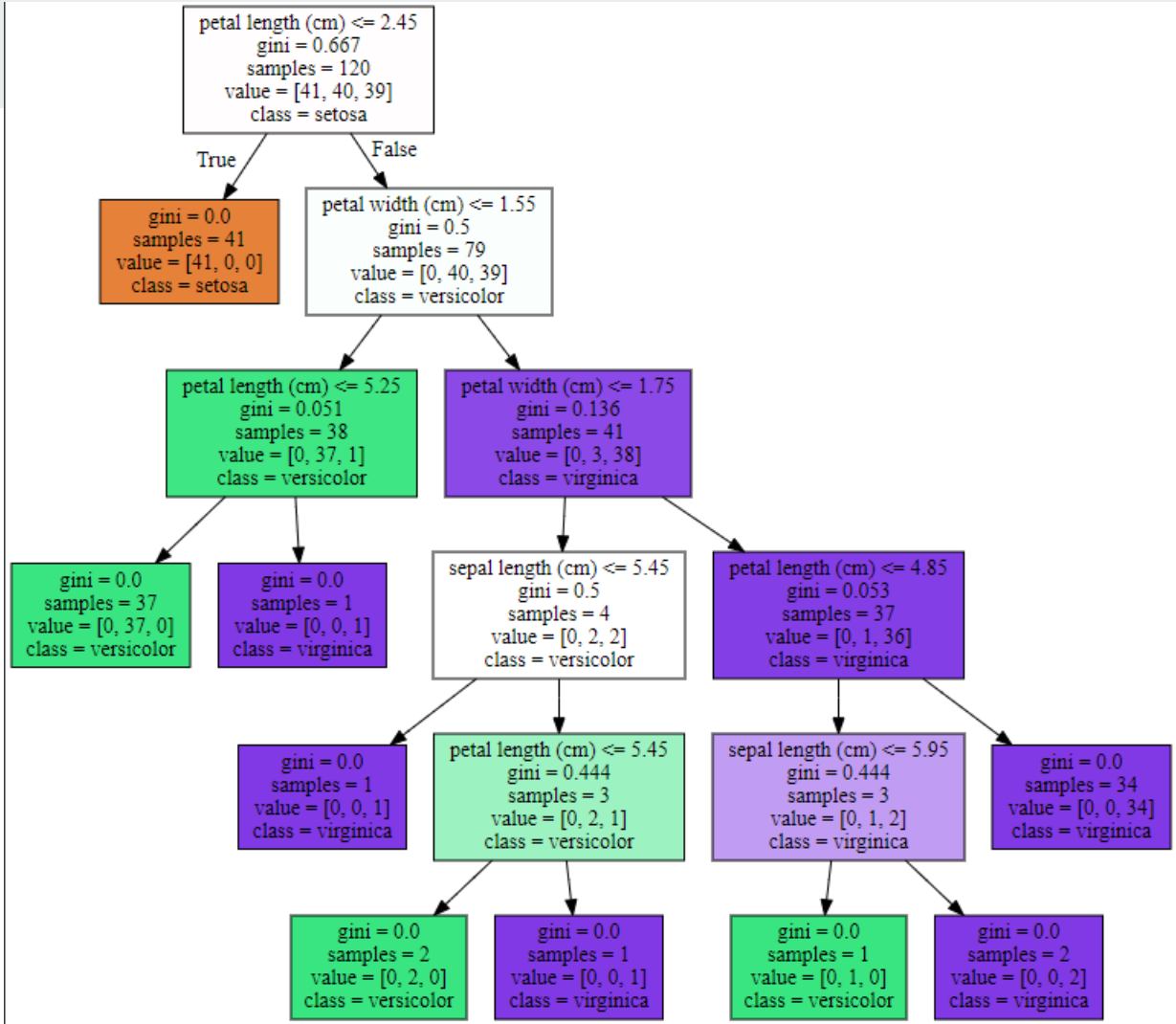


Target : setosa, versicolor, virginica

- petal length(cm) <2.45 : 피처의 조건이 있는 것은 자식 노드를 만들기 위한 규칙 조건, 없으면 leaf node
- gini : 다음의 value=[]로 주어진 데이터 분포에서의 지니 계수
- samples는 현 규칙에 해당하는 데이터 건수
- value=[] : 클래스 값 기반의 데이터 건수. 붓꽃 데이터세트는 클래스 값으로 0,1,2를 가지고 있으며, 0: Setosa, 1: Versicolor, 2: Virginica 품종을 가리킴.
- Class는 value 리스트 내에 가장 많은 건수를 가진 결정 값

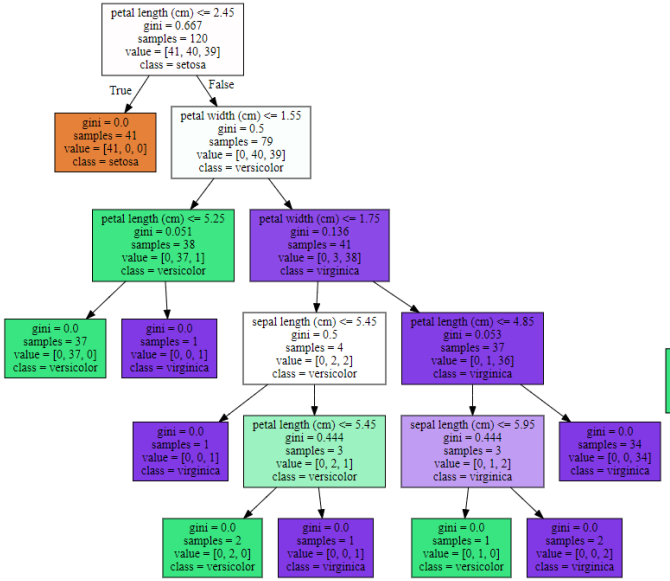
Graphiviz

- 결정트리 모델의 Graphiviz 시각화 결과

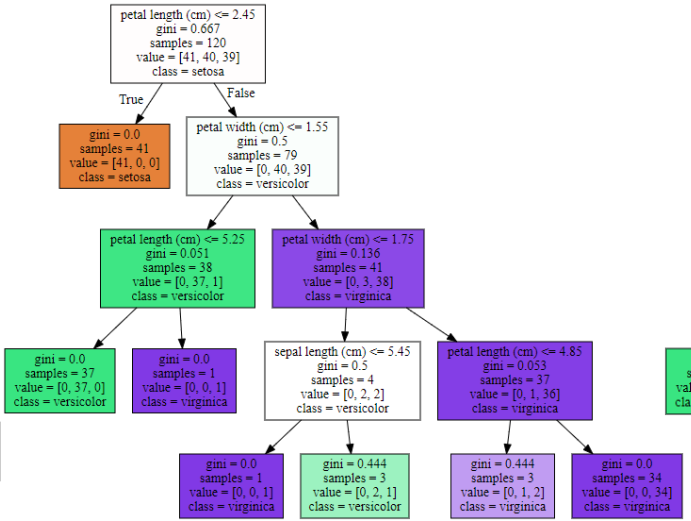


Max_depth에 따른 결정 트리 구조

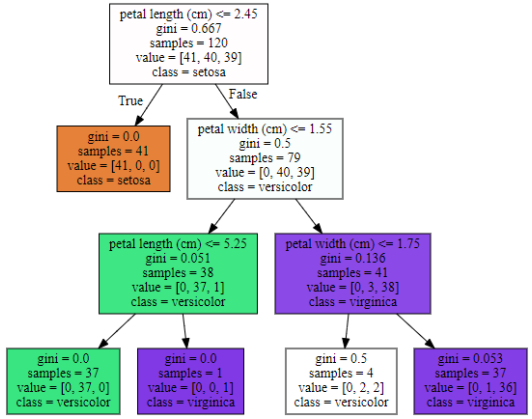
- Max_depth제약 없음



- Max_depth=4

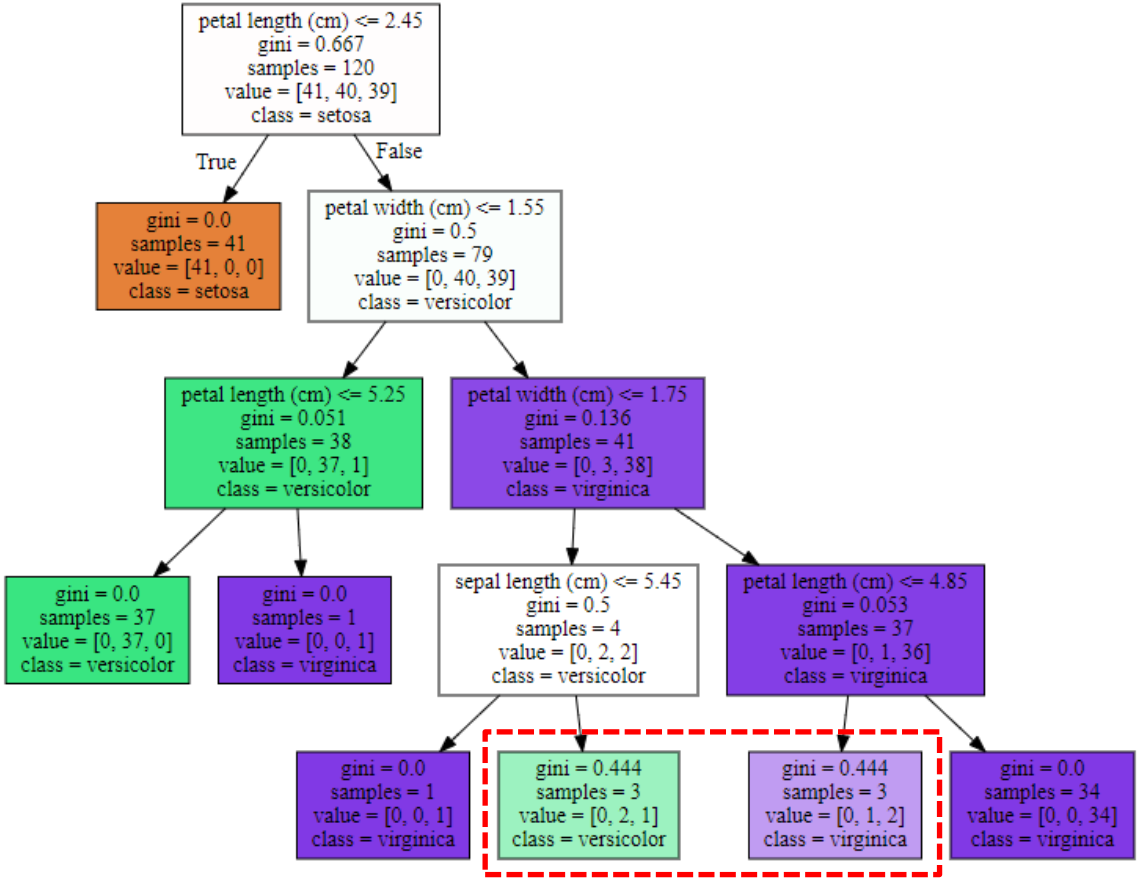


- Max_depth=3



Min_samples_split에 따른 결정 트리 구조

- min_samples_split=4
- 정확도 0.933

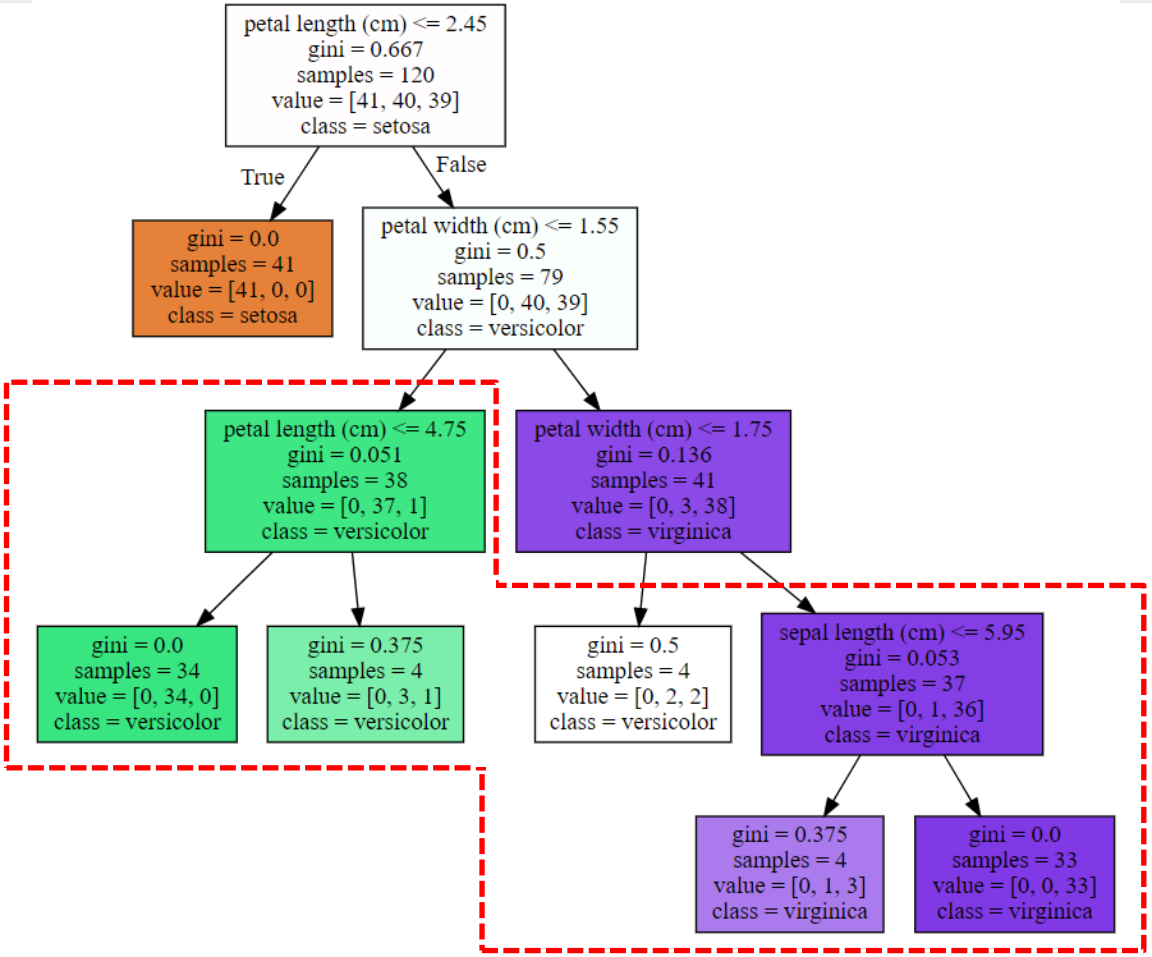


min_samples_split=4로 설정,
samples가 3개 이므로
서로 다른 class값이 있어도
split하지 않음.

Min_samples_leaf에 따른 결정트리 구조

- min_samples_leaf=4
- 정확도 0.933

min_samples_split=4이상인 노드는 리프 클래스 노드가 될 수 있으므로 규칙이 sample 4인 노드를 만들 수 있는 상황을 반영하여 변경됨



결정 트리의 feature 선택 중요도

- 사이킷런의 DecisionClassifier 객체는 feature_importances_을 통해 학습/예측을 위해서 중요한 feature들을 선택할 수 있게 정보를 제공함
- model_dtc.feature_importances_
- sns.barplot(x=model_dtc.feature_importances_, y=iris_data.feature_names