📌 4.1 분류(Classification)의 개요

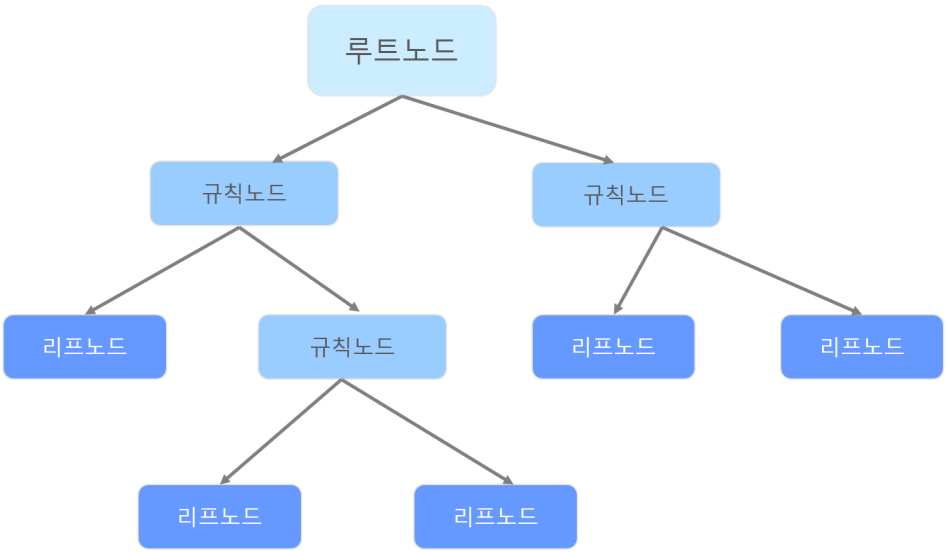
- 지도학습의 대표적 유형 중 하나
  지도학습: 명시적 정답(=레이블 값)이 있는 데이터가 주어진 상태에서 학습하는 머신러닝 방식
- 데이터의 피처와 레이블 값을 머신러닝 알고리즘으로 학습 후 모델 생성, 해당 모델로 새로운 데이터가 주어지면 해당 레이블 값을 예측!
  - 나이브 베이즈 : 베이즈 통계, 생성 모델 기반
  - 로지스틱 회귀 : 선형 관계성 기반 회귀 모델
  - 결정 트리 : 데이터 균일도에 따른 규칙 기반
  - 서포트 벡터 머신 : 클래스간 최대 구별 마진
  - 최소 근접 알고리즘 : 근접 거리 기준의 알고리즘 (ex. KNN)
  - 신경망 : 심층 연결 기반
  - 앙상블 : 여러 알고리즘 결합

📌 4.2 결정 트리

: 데이터에 있는 규칙을 학습을 통해 자동으로 찾아내서 트리 기반 분류 규칙 생성
쉽게 말해 if else의 연속!

트리 구조

- 규칙노드: 규칙조건을 나타냄, 해당 조건에서 뻗어나온 노드가 리프 또는 또다른 규칙노드가 되고 이들을 묶어서 서브트리라고 표현
- 리프노드: 결정된 클래스 값. 더 이상 분할되지 않음. (=더이상 자식이 없다!)



⭐규칙 노드가 많아짐 -> 분류 결정이 복잡해짐 -> !과적합! -> 성능 저하

균일도 측정

- 엔트로피를 이용한 정보 이득 지수 = 1 - 엔트로피지수(혼잡도)
- 지니계수: 0이 가장 평등, 1이 가장 불평등
- 정보이득이 높을수록, 지니계수가 낮을 수록 더 균일하다!
- DecisionTreeClassifier는 지니계수를 기반으로 데이터 분할

분할 방식

1. 해당 데이터가 모두 같은 분류에 속하는가?
2. T -> 리프노드로 만들어 분류 결정
   F -> 가장 좋은 분할 기준을 찾고 분할
3. 모든 데이터 집합의 분류가 결정될 때까지 위 과정 반복

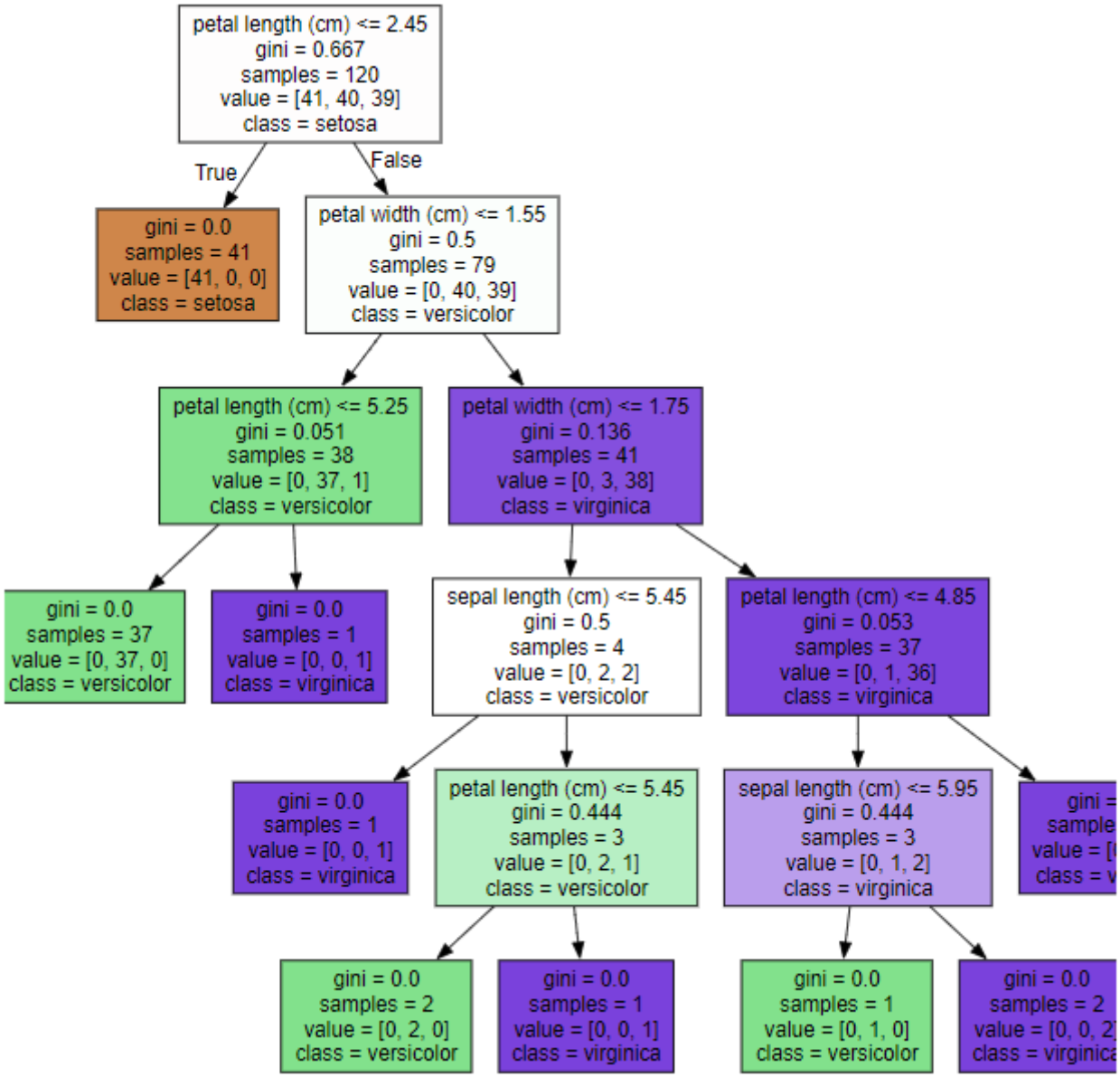
특징

+쉽고 직관적
+사전 가공(스케일링, 정규화 등)의 영향도가 낮음
-과적합으로 알고리즘 성능이 떨어질 수 있음!!

파라미터

- min_samples_split : 노드 분할을 위한 최소한의 샘플 데이터 수 (default : 2)
- min_samples_leaf : 분할 이후의 브랜치 노드에서 가져야할 최소 샘플 데이터 수
    - 비대칭적 데이터의 경우 특정 클래스의 데이터값이 극도로 작을 수 있음. 해당 조건에선 작게 설정해야함
- max_features : 분할을 위해 고려할 최대 피처 개수 (default : None -> 전체 피처 고려)
    - int형 : 대상 피처의 개수, float형 : 전체 피처 중 대상 피처의 퍼센트
    - sqrt : 전체 피처 개수의 제곱근만큼 선정
    - auto : 위의 sqrt와 동일
    - log : log2(전체피처개수)만큼 선정
- max_depth : 트리의 최대 깊이 규정 (default : None -> 가능한 한 완벽하게 클래스 결정값이 될때까지 분류)
- max_leaf_nodes : 말단 노드(=Leaf)의 최대 개수

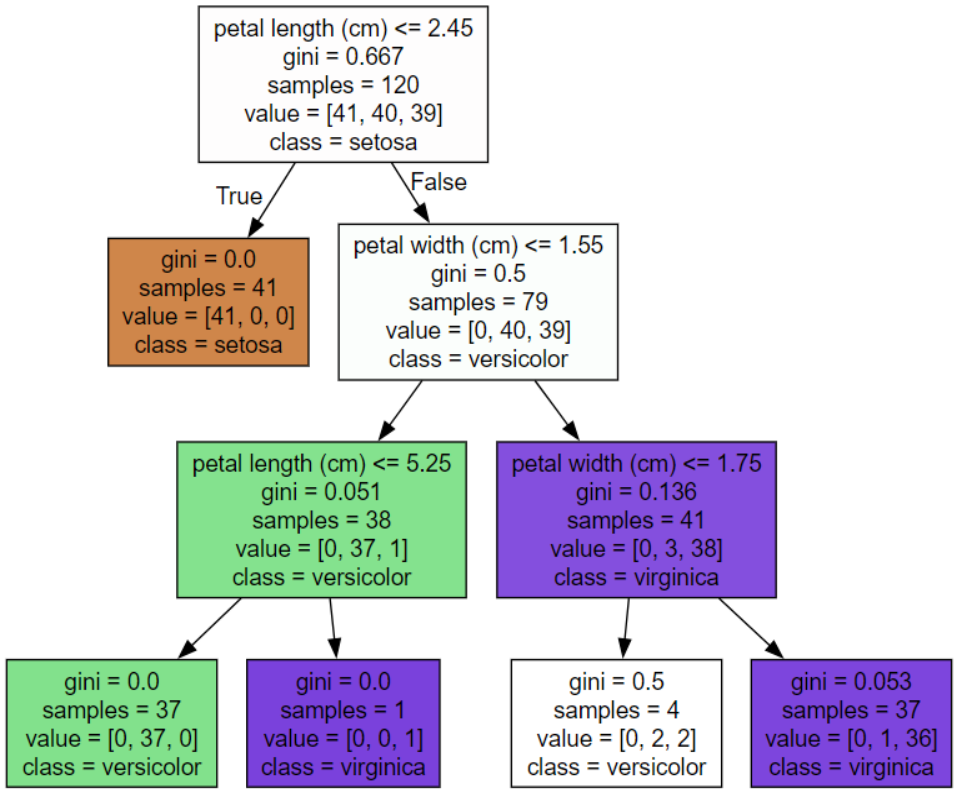
위의 파라미터들을 잘 조절해서 과적합이 일어나지 않도록 하는 것이 중요함



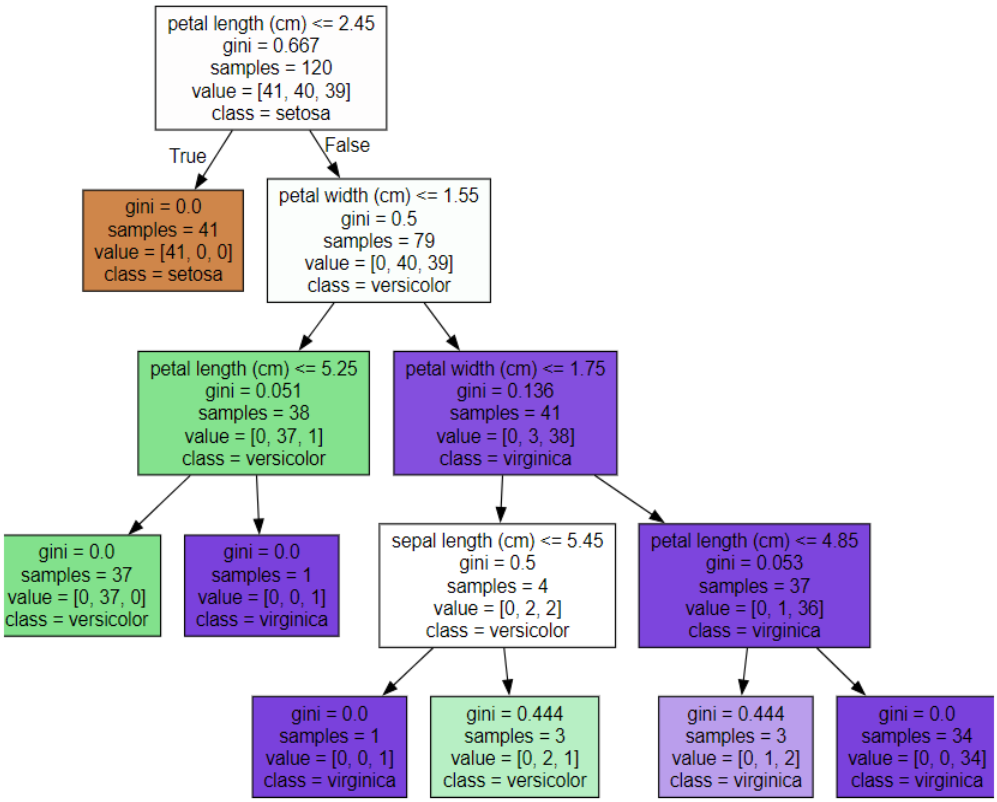
위의 iris 데이터를 기반으로 학습한 결정 트리로 더 자세히 살펴보기

- 규칙 노드의 가장 윗줄은 해당 노드를 나누는 조건(이게 없으면 리프 노드)
- gini는 데이터 분포에서의 지니 계수
- samples는 해당 노드, 해당 규칙에 해당하는 데이터 건수
- values = [] 은 클래스값 기반의 데이터 건수(아이리스의 경우 꽃 종류로 0, 1, 2)
- class = setosa 는 하위 노드를 가질 경우에 setosa의 개수가 가장 많다는 의미

- 색깔이 진할수록 지니계수가 낮고(= 균일하고) 해당 레이블에 속하는 데이터가 많다는 의미



- max_depth = 3으로 규정, 트리의 최대 깊이를 설정함
- 참고로 트리의 최대 깊이는 루트 노드를 제외한 노드들의 행 개수가 몇개인지!



- min_samples_split = 4로 샘플 개수가 4개 이상이 아니면 더 이상 나누지 않는 것을 확인 가능

📌 4.3 앙상블 학습

: 여러개의 분류기를 생성하고 그 예측을 결합 -> 보다 정확한 최종 예측 도출! 정형 데이터 분류에서 뛰어난 성능을 보여줌

보팅

: 서로 다른 알고리즘의 여러 분류기가 투표를 통해 최종 예측 결과 결정

- 하드 보팅
    - 다수결 원칙
    - 각 분류기에서 선정한 클래스 값 중 가장 많이 나온 클래스 값을 선정
- 소프트 보팅
    - 보통 선정되는 보팅 방식
    - 각 분류기에서 각 결과값의 확률을 구하고 그 평균을 구함 -> 해당 확률의 평균이 가장 높은 결과값이 선정

배깅

: 같은 유형의 알고리즘의 여러 분류기가 투표를 통해 최종 예측 결과 결정

- ex. 랜덤 포레스트 알고리즘
- 부스트래핑(Boostrapping): 개별 분류기에 데이터를 샘플링 해서 추출
    - 해당 방법으로 데이터 분할
    - 교차 검증과달린 데이터간의 중첩 허용

      부스팅
- 앞서 학습한 분류기의 예측이 틀린 데이터의 경우 올바르게 예측하도록 다음 분류기엔 가중치를 부여하며 학습
- ex. 그래디언트 부스트, XGBoost, LightGBM

📌 4.4 랜덤 포레스트

: 배깅 앙상블의 가장 대표적 알고리즘.

- 기반 알고리즘: 결정 트리
- 빠른 수행 속도, 높은 예측 성능
- 전체 데이터에서 일부가 중첩되게 샘플링된 데이터를 기반으로 학습

  파라미터
- n_estimators : 결정 트리 개수 지정 (default : 10)

- max_features : 결정 트리에 이용되는 것과 같음. 그러나 default가 auto로 전체 피처를 참조하는 게 아닌 전체 피처 수의 제곱근만큼을 참조
- 그 외 파라미터는 결정 트리에 사용되는 것과 같이 사용 가능