지도학습: 레이블, 즉 명시적인 정답이 있는 데이터가 주어진 상태에서 학습하는 머신러닝

* 분류: 머신러닝 알고리즘으로 학습데이터 학습해 모델 생성하고 미지의 레이블 값 예측
* 분류를 구현하는 알고리즘

1. 나이브 베이즈 Naïve Bayes: 베이즈 통계과 생성모델에 기반
2. 로지스틱 회귀 Logistic Regression: 독립변수와 종속변수의 선형 관계성에 기반
3. 결정 트리 Decision Tree: 데이터 균일도에 따른 규칙 기반

🡪 쉽고 유연, 데이터 스케일링이나 정규화 등 사전 가공의 영향이 적음

🡪 but 예측 성능 향상시키기 위해 복잡한 규칙 구조를 가져서 과적합 발생해 예측 성능 저하될 수 있음

1. 서포트 벡터 머신 SVM: 개별 클래스 간 최대 분류 마진을 효과적으로 찾아줌
2. 최소 근접 알고리즘 Nearest Neighbor: 근접 거리 기준으로
3. 신경망 Neural Network: 심층 연결 기반
4. 앙상블 Ensemble: 서로 다른(또는 같은) 머신러닝 알고리즘 결합

🡪 정형 데이터의 예측 분석 영역에서는 앙상블이 매우 높은 예측 성능 가짐

🡪 배깅(Bagging) 방식: 랜덤 포레스트 but 근래는 부스팅 방식으로 발전

🡪 부스팅(Boosting) 방식: 그래디언트 부스팅 – XgBoost, LightGBM

🡪 기본 알고리즘으로는 일반적으로 결정 트리 사용, 많은 여러 개의 약한 학습기를 결합해 확률적 보완과 오류 부분에 대한 가중치를 계속 업데이트하면서 예측 성능 향상 – 결정 트리가 좋은 약한 학습기임

**결정 트리**

* 데이터에 있는 규칙을 학습을 통해 자동으로 찾아내 트리 기반의 분류 규칙 만드는 것
* 어떤 기준을 바탕으로 규칙을 만들어야 가장 효율적인 분류가 될 것인지가 성능 좌우
* 새로운 규칙마다 서브 트리 생성됨, Leaf Node: 결정된 클래스 값, 규칙 노드: 규칙 조건이 됨
* 트리의 깊이가 깊어질수록 과적합으로 이어지기 쉬워 예측 성능 저하

🡪 가능한 적은 결정 노드로 높은 예측 정확도 가지려면 분류할 때 최대한 많은 데이터 세트가 해당 분류에 속할 수 있도록 규칙 정해져야 함

🡪 최대한 균일한 데이터 세트 구성하도록 분할

🡪 정보 균일도가 높은 데이터 세트 먼저 선택하도록 규칙조건 만듦, 서브 데이터에서 또 균일도 높은 자식 데이터 세트 쪼개는 방식으로

정보의 균일도 특정하는 방법: **엔트로피를 이용한 정보 이득 지수** 과 **지니 계수**

1. 정보 이득 지수 = 1 – 엔트로피 지수

엔트로피: 주어진 데이터 집합의 혼잡도, 서로 다른 값이 섞여 있으면 엔트로피 높음

**정보 이득이 높은 속성을 기준으로 분할**

1. 지니 계수: **지니 계수 낮을수록 데이터 균일도가 높음**

결정 트리 알고리즘을 구현한 DecisionTreeClassifier: 기본은 지니 계수 이용해 데이터 세트 분할

**결정 트리 모델 특징**

균일도를 기반으로 해서 알고리즘이 쉽고 직관적, 룰이 명확해 규칙 노드와 리프 노드가 어떻게 생성되는지 알 수 있고 시각화로 표현까지 가능, 피처의 스케일링과 정규화 같은 전처리 필요 X

But 과적합으로 정확도 떨어짐 – 피처가 많고 균일도가 다양할수록 트리 깊이 커지고 복잡

🡪 모든 상황 만족하는 완벽한 규칙 만들 수 없음. 트리의 크기를 사전에 제한하는 것이 성능 튜닝에 도움

**결정 트리 파라미터**

결정 트리 알고리즘 구현한 클래스: DecisionTreeClassifier(분류) DecisionTreeRegressor(회귀)

사이킷런의 결정 트리 구현은 CART 알고리즘 기반

DecisionTreeClassifier의 파라미터

1. **min\_samples\_split**: 노드를 분할하기 위한 최소한의 샘플 데이터 수, 디폴트 2, 과적합 제어

즉 자식 노드로 분할하려면 최소한 샘플 개수가 몇 개는 필요한데, 그보다 적으면 분할 X, 그냥 리프 노드가 됨.

1. **min\_samples\_leaf**: leaf node가 되기 위한 최소한의 샘플 데이터 수, 디폴트 1, 과적합 제어

디폴트 1 의미는 단독 클래스로만 되어 있거나 단 한 개의 데이터로 돼 있을 경우에는 리프 노드

비대칭적 데이터의 경우 특정 클래스의 데이터가 작을 수 있으므로 이때는 작게 설정

이 값을 키우면 리프 노드가 될 수 있는 조건이 완화됨, min\_samples\_leaf<=지정 값 만족하면 리프 노드 됨

1. **max\_features**: 최적의 분할 위해 고려할 최대 피처 개수, 디폴트 None(모든 피처 사용)

int면 대상 피처 개수, float면 전체 피처 중 대상 피처의 퍼센트, sqrt면 루트(전체 피처 개수),

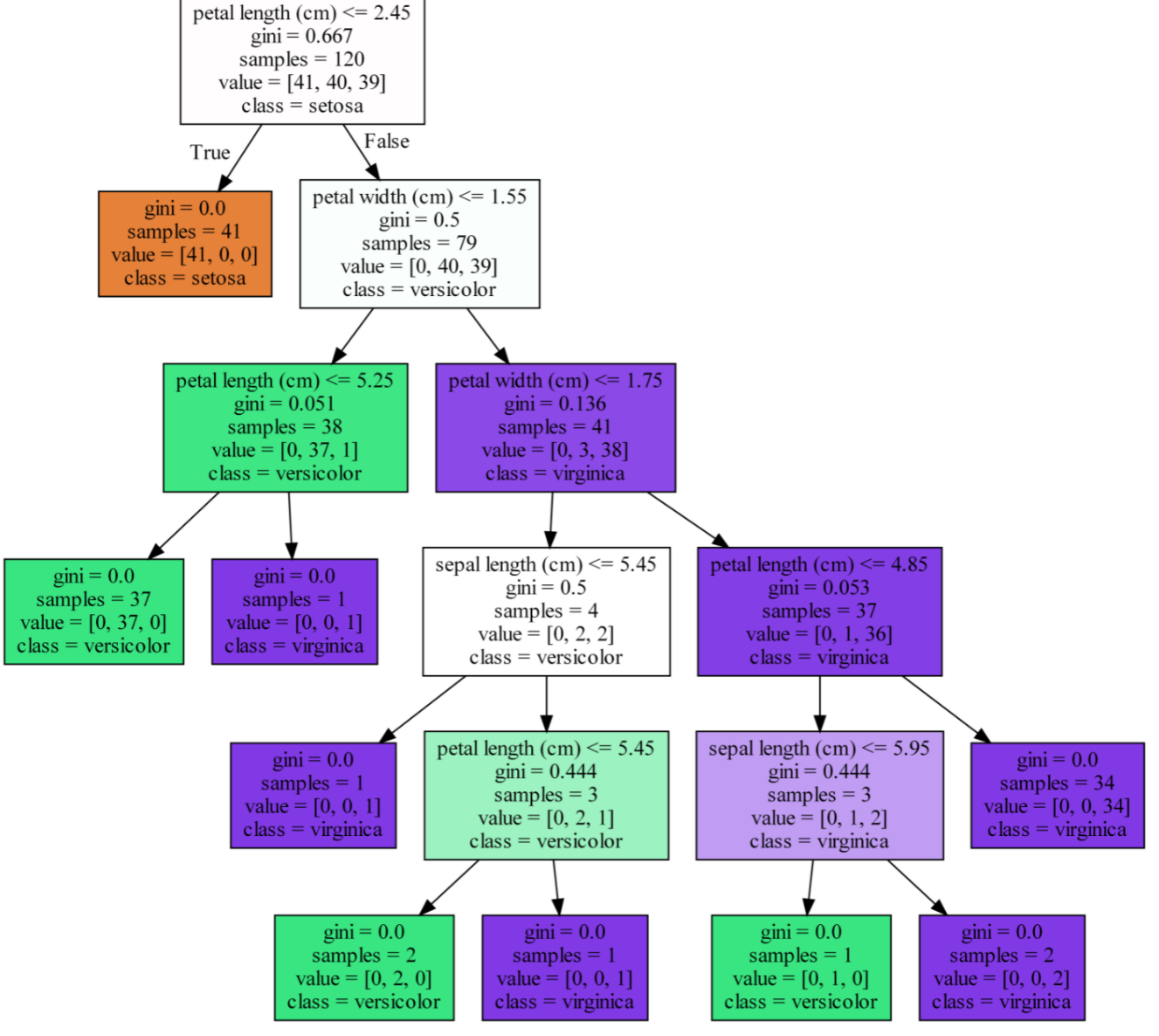
auto는 sqrt와 동일, log는 전체 피처 중 log2(전체 피처 개수) 선정

1. **max\_depth**: 최대 깊이, 디폴트 None,
2. **max\_leaf\_nodes**: leaf node의 최대 개수

**결정 트리 모델 시각화**

Graphviz - 결정 트리 알고리즘이 어떤 규칙을 가지고 트리 생성하는지 시각적으로 보여줌

export\_graphviz(학습 완료된 Estimator, 피처 이름 리스트, 레이블 이름 리스트) -> 학습된 결정 트리 규칙을 트리 형태로 시각화

리프 노드 – 자식 노드가 없는 노드, 최종 레이블 값이 결정되는 노드

브랜치 노드 – 자식 노드가 있는 노드, 자식 노드를 만들기 위한 분할 규칙 조건 가짐

Gini: value=[]으로 주어진 데이터 분포에서의 지니 계수

Samples는 현 규칙에 해당하는 데이터 건수, Class 는 하위 노드 가질 경우에 제일 많은 수의 레이블

Value = [] 는 클래스 값 기반의 데이터 건수 🡪 0: Setosa, 1: Versicolor, 2: Virginica 품종 가리킴

🡪 value=[41,40,39] 이면 0번 품종 41개, 1번 품종 40개, 2번 품종 39개라는 뜻

각 노드의 색깔은 붓꽃 데이터의 레이블 값 – 주황색 0: Setosa, 초록색1: Versicolor, 보라색2: Virginic 🡪 색깔이 짙어질수록 지니 계수 낮고 해당 레이블에 속하는 샘플 데이터가 많다.

규칙 생성 로직을 미리 제어하지 않으면 완벽하게 클래스 구별해내기 위해 트리 노드 계속 만듦 🡪 과적합 🡪 하이퍼 파라미터가 복잡한 트리 생성되는 것을 막음( max\_depth, min\_samples\_split, min\_samples\_leaf)

결정 트리는 균일도에 기반해 어떠한 속성을 규칙 조건으로 선택하느냐가 중요함

피처의 중요한 역할 지표를 DecisionTreeClassifier객체의 feature\_importances\_속성으로 제공

🡪 ndarray형태, 피처 순서대로 값 할당, 값이 높을수록 해당 피처의 중요도 높음