

트리 구조를 기반으로 데이터에서 규칙 학습하는 알고리즘 명: 결정트리

- 해결 가능한 문제: 분류, 회귀
- 복잡한 데이터에서도 ( 빠르게 ) 동작

결정트리 첫 기준: Root node

결정트리 중간 기준: Intermediate Node

마지막 기준: Leaf Node

결정트리는 다른 알고리즘에 비하여 전처리가 많이 필요하다 O, X

결정트리에서 과대적합 방지를 위하여: 가지치기

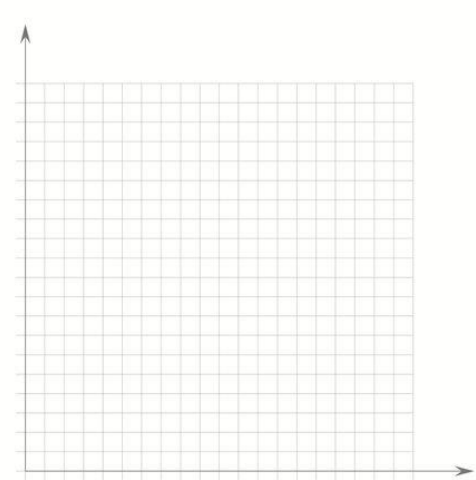
Parameter	Description
max_depth	트리의 최대 깊이 제한, 감소시킬수록 과대적합의 위험이 줄어든다.
max_features	각 노드에서 분할에 사용할 features 개수
min_sample_split	분할되기 위한 최소 샘플 수 조건
min_sample_leaf	리프 노드가 가져야할 최소 샘플 수
min_weight_fraction_leaf	min_sample_leaf 와 동일 (가중치가 부여된 전체 샘플에서의 비율)
max_leaf_node	리프 노드의 최대 수
min_impurity_decrease	분할로 얻어질 최소한의 불순도 감소량

지니 불순도에서 완벽한 분류: 0

지니 불순도에서 완벽한 랜덤: 0.5

엔트로피에서 완벽한 랜덤: 1

지니 불순도 및 엔트로피 그래프:



결정트리 단점 2개: 과적합에 쉽게 빠짐, 샘플 수 증가시 효율 감소, 데이터에 방향에 민감

화이트박스 모델 예: 결정트리

블랙박스 모델 예: 랜덤포레스트, 신경망

```
tree_clf1 = DecisionTreeClassifier(random_state=42)
tree_clf2 = DecisionTreeClassifier(min_samples_leaf=5, random_state=42)
tree_clf1.fit(X_moons, y_moons)
tree_clf2.fit(X_moons, y_moons)

tree_reg = DecisionTreeRegressor(max_depth=2, random_state=42)
tree_reg.fit(X_quad, y_quad)
```

다양한 모델을 합쳐 학습: 앙상블 학습

차원을 줄여 필요한 부분의 성능을 강화하는 방법: 차원축소, 주성분 분석

직접 투표 분류기 종류: 강한 학습기, 약한 학습기

- 참/거짓: 강한
- 확률: 약한

훈련 세트에서 중복을 허용하여 샘플링: 배깅

- 학습 되지 못한 데이터로 테스트: OOB 평가

중복 허용 없이 샘플링: 페이스팅

처음에는 과소적합, 오분류에 가중치를 강하게 주는 기법: 부스팅

- 대표적 1개: AdaBoost

결정 트리 쌓기: 랜덤 포레스트

1차원식 감소하는 법: 투영

말린거 펼치기: 매니폴드

모든 것은 차원 축소 가능하다 가정: 매니폴드 가정

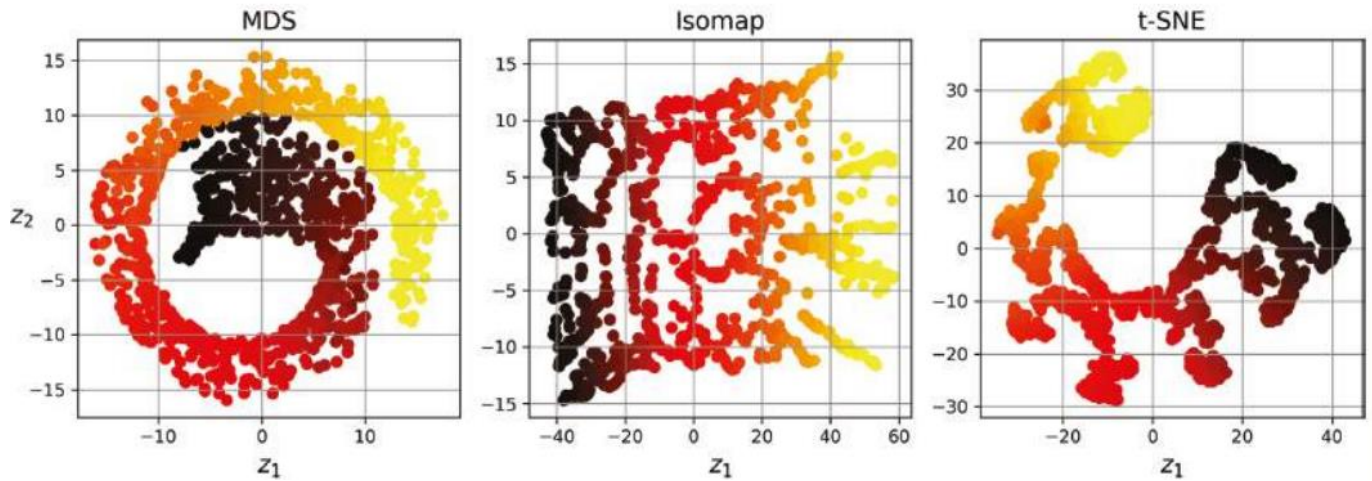
- 항상 유효한지: O, **X**

```
pca = PCA(n_components=2)
```

 n\_components의 뜻: 2차원으로 줄여라

```
pca = PCA(n_components=0.95)
```

 n\_components의 뜻: 손실을 5%만 허용하라



비지도 학습 종류 대표적 3개: 군집, 이상치 탐지, 밀도 추정

간단한 군집 알고리즘: k평균

- 샘플을 하나의 클러스터에 할당: 하드군집
- 클러스터마다 샘플에 점수 부여: 소프트 군집

이미지 분할 3종: 색상 분할, 시맨틱 분할, 인스턴스 분할

동일한 색상을 가진 픽셀을 같은 세그먼트에 할당: 색상 분할

동일한 종류의 물체에 속한 모든 픽셀을 같은 세그먼트에 할당: 시맨틱 분할

개별 객체에 속한 모든 픽셀을 같은 세그먼트에 할당: 인스턴스 분할

학습할 데이터를 전문가가 선택해서 학습하는 방식: 능동 학습

밀도기반 클러스터링: DBSCAN

```
dbscan = DBSCAN(eps=0.05, min_samples=5)
```

- eps역할: 한 클러스터의 범위
- min\_samples역할: 클러스터 안의 클러스터가 n개 있어야 유효한 클러스터

기타 군집 알고리즘

- 병합군집, BIRCH, 평균-이동, 유사도 전파, 스펙트럼 군집