Chapter 04. 분류

4.1 분류(Classification)의 개요

분류(Classification)

지도 학습의 대표적인 유형

학습 데이터로 주어진 데이터의 피처와 레이블값(결정 값, 클래스 값)을 머신러닝 알고리즘으로 학습해 모델을 생성하고, 생성된 모델에 새로운 데이터 값이 주어졌을 때 미지의 레이블 값을 예측하는 것

4.2 결정 트리

결정 트리(Decision Tree)

결정 트리: 데이터에 있는 규칙을 학습을 통해 자동으로 찾아내 트리 기반의 분류 규칙을 만드는 알고리즘
(if/else, 스무고개 게임)

규칙 노드: 규칙 조건, 리프 노드: 결정된 클래스 값, 서브 트리: 새로운 규칙 조건마다 생성되는 트리

많은 규칙 = 분류를 결정하는 방식이 복잡 ⇒ 과적합, 결정 트리의 예측 성능 저하

결정 노드는 정보 균일도가 높은 데이터 세트를 먼저 선택할 수 있도록 규칙 조건을 생성
정보의 균일도를 측정하는 대표적인 방법: 정보 이득 지수(높), 지니 계수(낮)

- 정보 이득 지수: 1 - 엔트로피 지수
  엔트로피: 주어진 데이터 집합의 혼잡도 / 서로 다른 값 → 높, 같은 값 → 낮
- 지니 계수: (경제학에서) 불평등 지수를 나타낼 때 사용하는 계수 / 평등 → 0, 불평등 → 1

결정 트리 모델의 특징

장점: 쉽고 직관적임, 피처의 스케일링이나 정규화 등의 사전 가공 영향도가 크지 않음.

단점: 과적합으로 알고리즘 성능이 떨어짐 → 트리의 크기를 사전에 제한하는 튜닝 필요

결정 트리 파라미터

사이킷런 결정 트리 구현은 CART(Classification And Regression Trees) 알고리즘 기반

- `DecisionTreeClassifier()`: 분류를 위한 클래스

min_samples_split	노드를 분할하기 위한 최소 샘플 데이터 수 (default = 2) 작게 설정할수록 분할되는 노드 증가 → 과적합 가능성 증가
min_samples_leaf	분할이 될 경우, 왼/오 브랜치 노드에서 가져야 할 최소 샘플 데이터 수 큰 값일수록 노드 분할을 상대적으로 덜 수행 특정 클래스의 데이터가 극도로 작은 비대칭 데이터의 경우 작게 설정
max_features	최대 피처 개수 (default = None)
max_depth	트리의 최대 깊이 (default = None) 깊이가 깊어지면 과적합 가능성 증가
max_leaf_nodes	말단 노드(Leaf)의 최대 개수

- `DecisionTreeRegressor()`: 회귀를 위한 클래스

결정 트리 모델의 시각화

Graphviz 패키지, 사이킷런 `export_graphviz()`

- petal length(cm) <= 2.45: 자식 노드를 만들기 위한 규칙 조건, 없으면 리프 노드
- gini: value=[ ] 로 주어진 데이터 분포에서의 지니 계수
- samples: 현 규칙에 해당하는 데이터 건수
- value = [ ]: 클래스 값 기반의 데이터 건수

사이킷런 DecisionTreeClassifier 객체의 `feature_importances_`

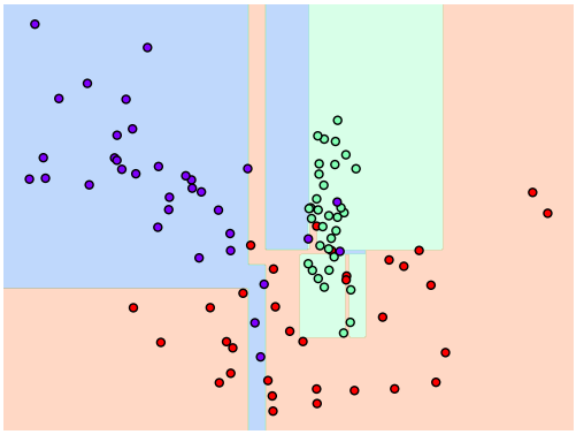
피처가 트리 분할 시 정도 이득이나 지니 계수를 얼마나 효율적으로 잘 개선시켰는지를 정규화한 값

일반적으로 높은 값 = 해당 피처의 중요도가 높음

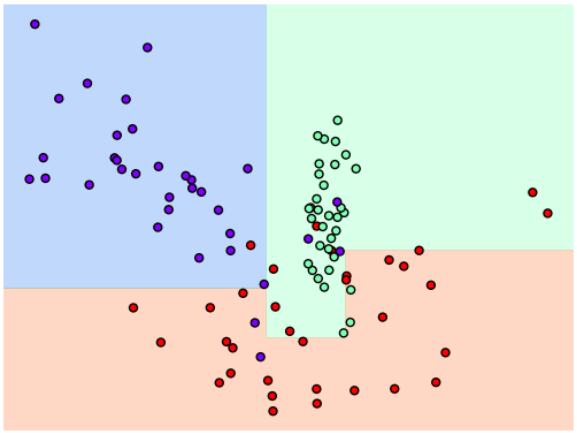
결정 트리 과적합(Overfitting)

`make_classification()`: 분류를 위한 테스트용 데이터 생성
피처 데이터 세트(X_features)와 클래스 레이블(y_labels) 데이터 세트가 반환됨

유틸리티 함수 visualize_boundary(): ML 모델이 클래스 값을 예측하는 결정 기준을 색상과 경계로 나타냄.



min_samples_leaf=1 (default)



min_samples_leaf=6

리프 노트 생성 규칙을 완화(min_samples_leaf=1 → min_samples_leaf=6)하여 이상치에 크게 반응하지 않으면서 좀 더 일반화된 분류 규칙에 따라 분류됨.

결정 트리 실습 - 사용자 행동 인식 데이터 세트

4.3 앙상블 학습

앙상블 학습 개요

앙상블 학습(Ensemble Learning)

여러 개의 분류기(Classifier)를 생성하고 그 예측을 결합함으로써 보다 정확한 최종 예측을 도출하는 기법

앙상블 학습 유형

- 보팅(Voting): 서로 다른 알고리즘을 가진 분류기를 결합
- 배깅(Bagging): 각각의 분류기가 모두 같은 유형의 알고리즘 기반. 데이터 샘플링을 서로 다르게 가져가면서 학습 수행 (ex. 랜덤 포레스트 알고리즘)
- 부스팅(Boosting): 여러 개의 분류기가 순차적으로 학습 수행. 앞에서 학습한 분류기가 예측이 틀린 데이터에 대해서 다음 분류기에 가중치(weight)를 부여하여 진행 (ex. 그래디언트 부스트, XGBoost, LightGBM)
- 스태깅(Stacking): 여러 가지 다른 모델의 예측 결괏값을 다시 학습 데이터로 만들어서 다른 모델로 재학습

보팅 유형 - 하드 보팅(Hard Voting)과 소프트 보팅(Soft Voting)

- 하드 보팅: 다수의 classifier 간 다수결로 최종 class 결정
- 소프트 보팅: 다수의 classifier들의 class 확률을 평균하여 결정

일반적으로 하드 보팅보다는 소프트 보팅이 예측 성능이 좋아서 더 많이 사용됨

보팅 분류기(Voting Classifier)

- `VotingClassifier()`: estimators - 리스트 값으로 보팅에 사용될 여러 개의 classifier 객체들을 튜플 형식으로 입력, voting - ‘hard’ 하드 보팅(default), ‘soft’ 소프트 보팅을 적용하라는 의미

4.4 랜덤 포레스트

랜덤 포레스트의 개요 및 실습

랜덤 포레스트

배깅의 대표적인 알고리즘.

여러 개의 결정 트리 분류기가 전체 데이터에서 배깅 방식으로 각자의 데이터를 샘플링해 개별적으로 학습을 수행한 뒤 최종적으로 모든 분류기가 보팅을 통해 예측을 결정

- 부트스트래핑 분할 방식을 이용하여 전체 데이터에서 일부가 중첩되게 데이터 샘플링
  부트스트래핑(bootstrapping): 여러 개의 데이터 세트를 중첩되게 분리하는 것
- 데이터 세트에 여러 개의 결정 트리 분류기를 각각 적용하여 개별적으로 학습 수행
- 보팅으로 최종 예측

랜덤 포레스트 하이퍼 파라미터 및 튜닝

사이킷런 `RandomForestClassifier()`

- `n_estimators`: 결정 트리의 개수, 서브 데이터세트 수 (default = 10)
- `max_features`: 최대 피처 개수 (default = auto, sqrt, 즉, 피처가 16개라면 4개만 참조)
- `max_depth`, `min_samples_leaf`, `min_samples_split` : 과적합을 개선하기 위해 사용