[Chapter 04. 분류]

1. 분류(Classification)의 개요

   1) 지도학습: 레이블(명시적 정답)이 있는 데이터가 주어진 상태에서 학습하는 머신러닝 방식
      기존 데이터 레이블의 패턴 → 알고리즘 인지 → 새로운 데이터 레이블 판별
   2) 분류를 구현하는 다양한 알고리즘
      Ex) 나이브 베이즈, 로지스틱 회귀, 결정 트리, 서포트 벡터 머신, 최소 근접 알고리즘, 신경망, 앙상블…
   3) 앙상블: 서로 다른/또는 같은 알고리즘 결합 (대부분 동일한 알고리즘 결합)
      (1) 배깅 방식
          Ex) 랜덤 포레스트
      (2) 부스팅 방식
          Ex) 그래디언트 부스팅, XGBoost, LightGBM, Stacking

2. 결정 트리(Decision Tree): 데이터에 있는 규칙을 학습을 통해 자동으로 찾아내 트리 기반의 분류 규칙 만들기
   1) 결정 트리의 구조
      (1) 규칙 노드: 규칙 조건
      (2) 리프 노드: 결정된 클래스 값
      (3) 서브 트리: 새로운 규칙 조건마다 생성됨

   2) 결정 트리 분할 – 정보의 균일도 측정
      (1) 정보 이득 지수: 엔트로피 개념을 기반으로 함 (엔트로피: 주어진 데이터 집합의 혼잡도), 1 – (엔트로피 지수)
      (2) 지니 계수: 불평등 지수, 낮을수록 데이터 균일도가 높음
   3) 결정 트리 모델의 특징
      (1) 장점: 알고리즘이 쉽고 직관적임, 전처리 작업 불필요, 가공 영향도 크지 않음

(2) 단점: 과적합으로 정확도 감소 → 트리 크기를 사전 제한하는 튜닝 필요

4) 결정 트리 파라미터

(1) min_samples_split: 과적합 제어용, 디폴트 = 2, 작게 설정할수록 분할되는 노드 많아져서 과적합 가능성 증가

(2) min_samples_leaf: 과적합 제어용 but 비대칭적 데이터는 특정 클래스의 데이터가 극도로 작을 수 있으므로 작게 설정 필요

(3) max_features: 최적의 분할을 위해 고려할 최대 피처 개수, 디폴트 = None, 데이터 세트의 모든 피처를 사용해 분할 수행

(4) max_depth: 트리의 최대 깊이 규정, default = None

(5) max_leaf_nodes: 말단 노드의 최대 개수

5) 결정 트리 모델의 시각화

Graphviz 사용

export_graphviz(): 그래프 형태로 시각화할 수 있는 출력 파일 생성

Ex) 붓꽃 데이터 세트

- 리프 노드 = 최종 클래스 값이 결정되는 노드
- 자식 노드가 있는 노드 = 브랜치 노드

fearute_importances_: ndarray 형태로 값을 반환, 피처 순서대로 값이 할당됨, 값이 높을수록 피처의 중요도가 높음

6) 결정 트리 과적합

visualize_boundary(): 머신러닝 모델이 클래스 값을 예측하는 결정 기준을 색상과 경계로 나타냄

학습 데이터에만 지나치게 최적화된 분류 기준은 오히려 테스트 데이터 세트에서 정확도를 떨어뜨릴 수 있음

7) 결정 트리 실습 – 사용자 행동 인식 데이터 세트

- 중복된 피처명에 대해서는 원본 피처병에 _1 또는 _2를 추가로 부여해 변경한 뒤 로드
- 결정 트리의 깊이가 예측 정확도에 주는 영향: 깊어진 트리는 학습 데이터 세트에는 올바른 예측 결과를 가져올지 모르나 검증 데이터 세트에서는 오히려 과적합으로 인한 성능 저하 유발 → 파라미터로 제어

3. 앙상블 학습

- 앙상블 학습: 여러 개의 분류기를 생성하고 그 예측을 결합함으로써 보다 정확한 최종 예측을 도출하는 기법 → 신뢰성 높은 예측값 get
- 비정형 데이터의 분류는 딥러닝이 뛰어난 성능을 보이나 대부분의 정형 데이터 분류 시에는 앙상블이 뛰어난 성능 나타냄

(1) 보팅 : 여러 개의 분류기가 투표를 통해 최종 예측 결과 결정 (서로 다른 알고리즘을 가진 분류기 결합)

① 하드 보팅 – 다수의 분류기가 결정한 예측값을 최종 보팅 결괏값으로 선정

② 소프트 보팅 – 레이블 값 결정 확률 모두 더하고 평균해서 확률이 가장 높은 레이블 값을 최종 보팅 결괏값으로 선정

보팅 분류기

VotingClassifier

주요 생성 인자로 estimators, voting 값 입력받음

default = hard

(2) 배깅 : 여러 개의 분류기가 투표를 통해 최종 예측 결과 결정 (각각의 분류기가 모두 같은 유형의 알고리즘 기반, 데이터 샘플링을 서로 다르게 가져감)

(3) 부스팅 : duf 개의 분류기가 순차적으로 학습을 수행하되, 앞에서 학습한 분류기가 예측에 틀린 데이터에 대해서는 올바르게 예측할 수 있도록 가중치 부여하며 학습, 예측 진행

(4) 스태킹: 여러 가지 다른 모델의 예측 결괏값을 다시 학습 데이터로 만들어서 다른 모델로 재학습시켜 결과 예측하는 방법

4. 랜덤 포레스트

1) 랜덤 포레스트의 개요 및 실습

- 배깅의 대표적인 알고리즘은 랜덤 포레스트
- 부트스트래핑: 여러 개의 데이터 세트를 중첩되게 분리하는 것
- 서브세트 데이터 임의로 만들어짐
- 랜덤 포레스트: 중첩된 개별 데이터 세트에 결정 트리 분류기를 각각 적용하는 것

2) 랜덤 포레스트 하이퍼 파라미터 및 튜닝

- 트리 기반 앙상블 알고리즘의 단점: 하이퍼 파라미터 너무 많음 → 튜닝하는데 시간이 많이 듦, 성능이 크게 향상되는 경우 적음
- GridSearchCV 적용하여 튜닝