2장. 사이킷런으로 시작하는 머신러닝

01. 사이킷런 소개와 특징

- 파이썬 머신러닝 라이브러리 중 가장 많이 사용됨
- 머신러닝을 위한 다양한 알고리즘 제공
- 개발을 위한 편리한 프레임워크와 API 제공

02. 첫번째 머신러닝 만들어 보기 - 붓꽃 품종 예측하기

- 지도 학습
    - 학습을 위한 다양한 피처와 레이블 데이터로 모델을 학습한 뒤, 별도의 테스트 데이터 세트로 미지의 레이블 예측
    - 학습 데이터 세트: 학습을 위해 주어진 데이터 세트
    - 테스터 데이터 세트: 머신러닝 모델의 예측 성능을 평가하기 위해 별도로 주어진 데이터 세트
    - 학습 데이터와 테스트 데이터는 반드시 분리해야 함 - test_split() API 이용
    - 방법 중 하나로 분류(Classification)을 들 수 있음
- 붓꽃 품종 예측
    - 데이터 세트 생성: load_iris() 함수 이용
    - 데이터 세트 분리: 학습 데이터와 테스트 데이터로 분리
    - 모델 학습: ML 알고리즘, 의사 결정 트리
    - 예측 수행: 학습 데이터 기반으로 학습 후, 다른 데이터(일반적으로 테스트 데이터 세트)를 이용해 예측
    - 평가: 예측 정확도 측정을 위해 accuracy_score() 함수 이용

03. 사이킷런의 기반 프레임 워크 익히기

- Estimator 이해 및 fit(), predict() 메서드

    - 분류 알고리즘 구현 클래스(Classifier)와 회귀 알고리즘 구현 클래스(Regressor)를 통칭
    - fit()과 predict()를 내부에서 구현
    - 지도학습과 달리 비지도학습에서는 fit()으로 변환을 위한 사전 구조를 맞추고, 실제 작업은 transform()으로 수행
    - fit()과 transform()을 하나로 결합한 fit_transform()도 제공
- 머신러닝 모델 구축을 위한 주요 프로세스
    - 피처 처리: 피처의 가공, 변경, 추출 수행
    - ML 알고리즘 학습/예측
    - 모델 평가
    - 사이킷런 패키지는 주요 프로세스를 지원하기 위한 다양한 모듈을 지원함
- 내장된 예제 데이터 세트
    - 예제용도 데이터 세트: 분류, 회귀 연습용
        - data: 피처의 데이터 세트, 넘파이 배열 타입
        - target: 분류 - 레이블 값 / 회귀 - 숫자 결괏값 데이터 세트, 넘파이 배열 타입
        - target_names: 개별 레이블 이름, 넘파이 or 파이썬 리스트 타입
        - feature_names: 피처의 이름, 넘파이 or 파이썬 리스트 타입
        - DESCR: 데이터 세트 및 각 피처에 대한 설명, 넘파이 or 파이썬 리스트 타입
    - 표본 데이터로 생성될 수 있는 데이터 세트: 분류, 클러스터링

04. Model Selection 모듈

- 학습/테스트 데이터 세트 분리 - train_test_split()
    - 학습 데이터와 테스트 데이터 세트를 분리해 예측을 수행해야 함
    - train_test_split()을 통해 원본 데이터 세트에서 학습 및 테스트 데이터 세트를 쉽게 분리 가능

- 교차 검증
    - 학습 또는 특정한 테스트 데이터 세트에만 과적합되는 모델이 만들어져 다른 데이터세트가 들어올 경우 성능이 저하되는 문제점을 해결하기 위해, 교차 검증을 이용하여 더 다양한 학습과 평가 수행
    - 여러 세트로 구성된 학습 데이터 세트와 검증 데이터 세트에서 학습과 평가 수행
    - K 폴드 교차 검증: K개의 데이터 폴드 세트를 만들어 K번만큼 학습과 검증 평가 반복적으로 수행
    - Statified K 폴드: 피처, 레이블 데이터 세트를 모두 이용해 불균형한 분포도를 가진 레이블 데이터를 다룰 때 사용
    - cross_val_score() : 폴드 세트 설정, 학습/테스트 데이터 추출, 학습/예측 수행하는 일련의 과정을 한꺼번에 수행
- GridSearchCV - 교차 검증과 최적 하이퍼 파라미터 튜닝을 한 번에
    - 하이퍼 파라미터: 머신러닝 알고리즘을 구성하는 주요 구성 요소
    - 하이퍼 파라미터를 순차적으로 입력하며 푀적의 파라미터를 도출할 수 있는 방안 제공

05. 데이터 전처리

- 기본 사항
    - 결손값(NaN, Null) 허용되지 않음
    - 문자열 값을 입력값으로 허용하지 않음
- 데이터 인코딩
    - 레이블 인코딩: 카테고리 피처를 코드형 숫자 값으로 변환, LabelEncoder를 객체로 생성한 후 fit()과 transform()을 호출해 레이블 인코딩 수행
    - 원-핫 인코딩: 행 형태의 피처 고유 값을 열 형태로 변환한 뒤, 고유 값에 해당하는 칼럼에만 1 표시, 나머지는 0
- 피처 스케일링과 정규화

    - 서로 다른 변수의 값 범위를 일정한 수준으로 맞춤
    - 표준화: 개별 피처를 평균이 0이고 분산이 1인 값으로 변환
    - 정규화(사이컷런에서 의미하는): 개별 벡터를 모든 피처 벡터의 크기로 나눔
- StandardScaler
    - 표준화를 쉽게 지원하기 위한 클래스
    - 개별 피처를 평균이 0이고, 분산이 1인 값으로 변환
- MinMaxScaler
    - 데이터값을 0과 1 사이의 범위 값으로 변환
- 학습 데이터와 테스트 데이터의 스케일링 변환 시 유의점
    - 가능하다면 전체 데이터의 스케일링 변환 후 학습/테스트 데이터로 분리
    - 여의치 않는 경우, 학습 데이터로 fit()이 적용된 스케일링 기준 정보를 그대로 테스트에 적용

06. 사이킷런으로 수행하는 타이타닉 생존자 예측

3장. 평가

01. 정확도

- 개념
    - 실제 데이터에서 예측 데이터가 얼마나 같은지 판단하는 지표
    - (예측 결과가 동일한 데이터 건수) / (전체 예측 데이터 건수)
- 주의사항
    - 단순한 알고리즘으로 예측 하더라도 데이터 구성에 따라 높은 정확도가 나올 수 있으니, 사용에 주의해야 함
    - 불균형한 레이블 값 분포에서 사용하기에 적합한 평가 지표 아님

02. 오차 행렬

- 개념
    - 이진 분류의 예측 오류가 얼마인지, 어떠한 유형의 예측 오류가 발생하고 있는지 나타내는 지표
    - 4분면 행렬에서 실제 레이블 클래스 값과 예측 레이블 클래스 값이 어떠한 유형을 가지고 매핑되는지 나타냄
    - 사이킷런에서 confusion_matrix() API 제공

TN(True Negative) : Negative 0으로 예측, 실제도 Negative 0	FP(False Positive) : Positive 1로 예측, 실제는 Negative 0
FN(False Negative) : Negative 0으로 예측, 실제는 Positive 1	TP(True Positive) : Positive 1로 예측, 실제도 Positive 1

03. 정밀도와 재현율

- 개념
    - positive 데이터 세트의 예측 성능에 좀 더 초점을 맞춘 평가 지표
    - 정밀도(양성 예측도) = TP / (FP + TP) : Positive으로 예측한 대상 중 Positive 한 데이터 비율

    - FP를 낮추는 데 초점
  - 재현율(민감도, TPR) = TP / (FN + TP) : Positive인 대상 중 Positive으로 예측한 데이터 비율
    -
    - FN을 낮추는 데 초점
- 정밀도/재현율 트레이드 오프
  - 정밀도와 재현율은 상호 보완적인 평가 지표여서 어느 한 쪽을 강제로 높이면 다른 한 쪽의 수치는 떨어지기 쉬움
  - 임곗값 증가 시 정밀도 값 높아짐, 감소 시 재현율 값 높아짐
- 맹점
  - 임곗값의 변경은 두 개의 수치를 상호 보완할 수 있는 수준에서 적용해야 함

04. F1 스코어

- 개념
  - 정밀도와 재현율을 결합한 지표
  - 정밀도와 재현율 어느 한쪽으로 찌우지지 않는 수치를 나타낼 때 상대적으로 높은 값을 가짐
  - f1_score() API 이용

05. ROC 곡선과 AUC

- 개념
  - 이진 분류의 예측 성능 측정
  - FPR(False Positive Rate)이 변할 때 TPR(True Positve Rate, 재현율)이 어떻게 변하는지
  - 직선에서 멀어질 수록 성능이 뛰어난 것
  - roc_curve() API 이용

06. 피마 인디언 당뇨병 예측