파이썬 기반의 머신러닝과 생태계 이해

머신러닝:

- 애플리케이션을 수정하지 않고도 데이터를 기반으로 패턴을 학습하고 결과를 예측하는 알고리즘을 통칭
- 데이터를 기반으로 숨겨진 패턴을 인지해 문제 해결
- 데이터에 매우 의존적

파이썬 기반의 머신러닝을 익히기 위해 필요한 패키지

- 머신러닝 패키지: 사이킷런; 데이터 마이닝 기반의 대표적인 머신러닝 패키지
- 행렬/선형대수/통계 패키지: 넘파이; 행렬과 선형대수를 다루는 패키지 + 사이파이
- 데이터 핸들링: 판다스; 대표적인 데이터 처리 패키지, 2차원 데이터 처리에 특화
- 시각화: 맷플롯립; 대표적 시각화 라이브러리, 너무 세분화되어 익히기 어려움 → 시본; 맷플롯립 기반의 보완된 시각화 패키지

넘파이

- 루프를 사용하지 않고 대량 데이터의 배열 연산 가능
- 대량 데이터 기반의 과학과 공학 프로그램은 빠른 계산 능력이 매우 중요 → 넘파이에 의존
- 넘파이는 매우 빠른 배열 연산을 보장하지만, 파이썬 언어 자체의 수행 성능 제약으로 수행 성능이 매우 중요한 부분은 C/C++ 기반의 코드로 작성하고 이를 넘파이에서 호출하는 방식으로 통합 가능

ndarray

- Import numpy as np : 넘파이 모듈 임포트
- array() : 인자를 ndarray로 변환
- shape 변수로 ndarray의 크기, 행과 열의 수를 튜플 형태로 가짐, ndarray 배열의 차원을 알 수 있음
- 데이터값으로는 서로 동일하나 차원이 달라서 오류가 생기는 경우가 빈번하므로 명확히 차수를 변환할 수 있어야 함

ndarray의 데이터 타입

- ndarray의 데이터값은 모두 가능
- ndarray내의 데이터 타입은 연산 특성상 같은 데이터 타입만 가능함
- ndarray내의 데이터 타입은 dtype 속성으로 확인 가능
- 리스트는 서로 다른 데이터 타입을 가질 수 있음
- 만약 다른 데이터 유형이 섞인 리스트를 ndarray로 변경하면 데이터 크기가 더 큰 데이터 타입으로 형 변환을 일괄 적용함
- astype() : ndarray 내 데이터 값의 타입 변경
- arange(), zeros(), ones() : ndarray 편리하게 생성하기
- ndarray는 tolist() 메서드를 이용해 리스트 자료형으로 변환 가능

인덱싱

- 특정한 데이터만 추출
  - axis 0이 로우 방향 축, axis 1이 칼럼 방향의 축 → 다차원 ndarray의 경우 축에 따른 연산 지원

정렬

- np.sort(), ndarray.sort(), argsort()
- np.sort()의 경우 원행렬은 그대로 유지한 채 원 행렬의 정렬된 행렬을 반환하며 ndarray.sort()의 경우 원행렬 자체를 정렬한 형태로 변환하며 반환 값은 None
- np.argsort() : 정렬 행렬의 원본 행렬 인덱스를 ndarray형으로 반환

선형대수 연산

- 행렬내적(행렬곱) : np.dot()
- 전치행렬 : np.transpose()

데이터 핸들링 – 판다스

- 2차원 데이터를 효율적으로 가공/처리할 수 있음
- 판다스의 핵심 객체는 DataFrame ; 여러 개의 행과 열로 이뤄진 2차원 데이터를 담는 데이터 구조체
- read_csv(filepath or buffer, sep = ' ') : 인자인 sep에 구분 문자를 입력하면 어떤 필드 구분 문자 기반의 파일 포맷도 DataFrame으로 변환이 가능

- read_fwf() : Fixed Width 기반이 칼럼 포맷을 DataFrame으로 로딩하기 위한 API

사이킷런으로 시작하는 머신러닝

Model Selection 모듈 소개

- 교차 검증
  - K 폴드 교차 검증: K개이 데이터 폴드 세트를 만들어서 K번만큼 각 폴드 세트에 학습과 검증 평가를 반복적으로 수행
  - Stratified K 폴드: 불균형한 분포도를 가진 레이블 데이터 집합을 위한 K폴드 방식
  - croess_val_score(): 폴드 세트 설정 → for 루프에서 반복으로 학습 및 테스트 데이터의 인덱스를 추출 → 반복적으로 학습과 예측을 수행, 예측 성능 반환

평가

분류의 성능 평가 지표: 정확도, 오차행렬, 정밀도, 재현율, F1 스코어, ROC AUC

정확도

- 예측 결과가 동일한 데이터 건수 / 전체 예측 데이터 건수

오차 행렬

- 학습된 분류 모델이 예측을 수행하면서 얼마나 헷갈리고 있는지도 함께 보여줌
- 이진 분류의 예측 오류가 얼마인지와 더불어 어떠한 유형의 예측 오류가 발생하고 있는지

정밀도와 재현율

- Positive 데이터 세트의 예측 성능에 초점을 맞춘 평가 지표
- 정밀도
  - TP / (FP + TP)
  - 실제 Negative 음성인 데이터 예측을 Positive 양성으로 잘못 판단하게 되면 업무상

큰 영향이 발생하게 되는 경우

- 재현율
  - TP / (FN + TP)
  - 실제 Positive 양성인 데이터 예측을 Negative로 잘못 판단하게 되면 업무 상 큰 영향이 발생하는 경우
- 분류하려는 업무의 특성상 정밀도/재현율이 특별히 강조돼야 할 경우 분류의 결정 임곗값을 조정해 정밀도/재현율의 수치를 높일 수 있음
- Positive 예측의 임곗값을 변경함에 따라 정밀도와 재현율이 수치가 변경됨 → 임곗값의 변경은 두개의 수치를 상호 보완할 수 있는 수준에서 적용되야 함

F1 스코어

- 정밀도와 재현율을 결함 한 지표
- 정밀도와 재현율이 아느 한쪽으로 치우치지 않는 수치를 나타낼 때 상대적으로 높은 값을 가짐
- $F1 = \frac{2}{\frac{1}{recall}+\frac{1}{precision}} = 2 * \frac{precision*recall}{precision+recall}$

ROC 곡선과 AUC

- ROC 곡선(수신자 판단 곡선): 렊이 변할 때 쎆이 어떻게 변하는지를 나타내는 곡선, FPR을 X축으로 TPR을 Y축으로 잡으면 FPR의 변화에 따른 TPR의 변화가 곡선 형태로 나타남
- FPR = FP / (FP + TN) = 1 – TNR = 1 - 특이성