<CHAPTER 1>

- Numpy

파이썬에서 선형대수 기반의 프로그램을 쉽게 만들 수 있도록 지원하는 패키지

- 데이터 타입

  숫자, 문자열, 불 들등 모두 가능

  Ndarray내의 데이터 타입은 같은 데이터 타입만 가능

- 인덱싱

  특정한 데이터만 추출: 원하는 위치의 인덱스 값을 지정하면 해당 위치의 데이터 반환

  슬라이싱: 연속된 인덱스상의 ndarray를 추출하는 방식. ':' 기호 사이에 시작 인덱스와 종료 인덱스를 표시하면 시작 인덱스에서 종료 인덱스 -1 위치에 있는 데이터의 ndarray를 반환함

  팬시 인덱싱: 일정한 인덱싱 집합을 리스트 또는 ndarray 형태로 지정해 해당 위치에 있는 데이터의 ndarray를 반환

  불린 인덱싱: 특정 조건에 해당하는지 여부인 True/False 값 인덱싱 집합을 기반으로 True에 해당하는 인덱스 위치에 있는 데이터의 ndarray를 반환

- Pandas

파이썬에서 데이터 처리를 위해 존재하는 가장 인기 있는 라이브러리

행과 열로 이뤄진 2차원 데이터를 효율적으로 가공/처리할 수 있는 다양한 기능 제공

넘파이와 상호 간의 변환이 빈번하기 발생함

- DataFrame의 [] 연산자

[]를 이용해서 원하는 칼럼의 데이터를 가져올 수 있음

Ex. Titanic_df의 처음 2개의 데이터 추출 -> titanic_df[0:2]

- DataFrame의 iloc[] 연산자

위치 기반 인덱싱 방식으로 동작

행과 열의 좌표 위치에 해당하는 정수 값을 입력해서 원하는 데이터를 추출할 수 있음

- DataFrame의 loc[] 연산자

명칭 기반으로 데이터를 추출

행 위치에는 DataFrame 인덱스 값을, 그리고 열 위치에는 칼럼명을 입력

<CHAPTER 2>

- 학습/테스트 데이터 세트 분리

테스트 데이터 세트를 이용하지 않고 학습 데이터 세트로만 학습하고 예측하면?

➔ 정확도가 100%로 나옴

이미 학습한 데이터 세트를 기반으로 예측했기 때문에 발생하는 문제

따라서 예측을 수행하는 데이터 세트는 학습을 수행한 학습용 데이터 세트가 아닌 전용의 테스트 데이터 세트여야 함.

사이킷런의 train_test_split()를 통해 원본 데이터 세트에서 학습 및 테스트 데이터 세트를 쉽게 분리할 수 있다.

- 교차검증

위의 학습/테스트 데이터 세트 분리 방법도 과적합에 취약한 약점을 가질 수 있음

(과적합=학습 데이터에만 과도하게 최적화되어 다른 데이터로 수행하는 경우 성능이 떨어지는 것)

교차 검증은 데이터 편중을 막기 위해 별도의 여러 세트로 구성된 학습 데이터 세트와 검증 데이터 세트에서 학습과 평가를 수행하는 것

- K 폴드 교차 검증

  K개의 데이터 폴드 세트를 만들어서 K번만큼 각 폴트 세트에 학습과 검증 평가를 반복적으로 수행하는 방법

- Stratified K 폴드

  불균형한 분포도 (특정 레이블 값이 특이하게 많거나 매우 적어서 값의 분포가 한쪽으로

치우치는 것)를 가진 레이블 데이터 집합을 위한 K 폴드 방식

원본의 데이터의 레이블 분포를 먼저 고려한 뒤 이 분포와 동일하게 학습과 검증 데이터 세트를 분배한다

일반적으로 분류에서의 교차 검증은 K 폴드가 아니라 Stratified K 폴드로 분할돼야 하고, 회귀에서는 Stratified K 폴드가 지원되지 않는다

- 데이터 전처리

1. 데이터 인코딩

   - 레이블 인코딩

   카테고리 피처를 코드형 숫자 값으로 변환하는 것

   몇몇 ML 알고리즘에서 변환된 숫자 값에 따라 순서나 중요도로 인식되는 문제점 발생

   - 원-핫 인코딩

   레이블 인코딩의 문제점을 위한 방식

   피처 값의 유형에 따라 새로운 피처를 추가해 고유 값에 해당하는 칼럼에만 1을 표시하고 나머지 칼럼에는 0을 표시함

2. 피처 스케일링과 정규화

   - 표준화

   데이터 피처 각각이 평균이 0이고 분산이 1인 가우시안 정규분포를 가진 값으로 변환

   개별 데이터를 모두 동일한 크기 단위로 비교하기 위해 최소 0 ~ 최대 1의 값으로 변환하는 것

   - 정규화

   개별 벡터의 크기를 맞추기 위해 변환하는 것

   개별 벡터를 모든 피처 벡터의 크기로 나눠 줌

   - MinMaxScaler

   데이터 값을 0과 1 사이의 범위의 값으로 변환

   데이터의 분포가 가우시안 분포가 아닐 때 적용할 수 있음

<CHAPTER 3>

- 정확도

= 예측 결과가 동일한 데이터 건수 / 전체 예측 데이터 건수

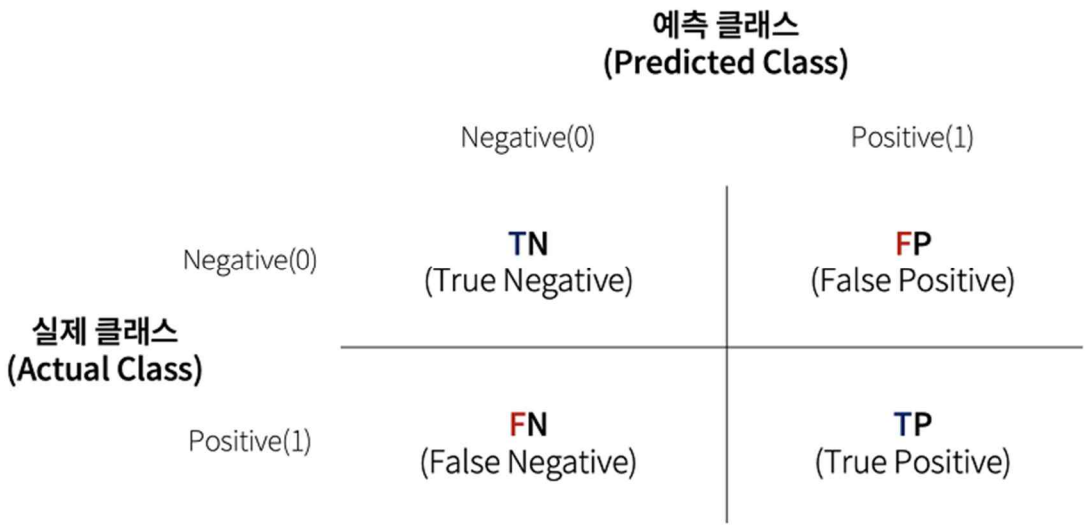
실제 데이터에서 예측 데이터가 얼마나 같은지를 판단하는 지표

직관적으로 모델 예측 성능을 나타내지만, 이진 분류의 경우 성능을 왜곡할 수 있음

- 오차행렬

정확도가 가지는 분류 평가 지표로서 한계점을 극복하기 위해 함께 쓰는 지표

이진 분류에서 성능 지표로 잘 활용되며, 학습된 분류 모델이 예측을 수행하면서 얼마나 헷갈리고 있는지 함께 보여줌



TN = 예측을 negative 값으로 했는데, 실제로도 negative

FP = 예측을 positive 값으로 했는데, 실제로는 negative

FN = 예측을 negative 값으로 했는데, 실제로는 positive

TP = 예측을 positive 값으로 했는데, 실제로도 positive

➔ 즉, 오차 행렬에서 정확도 = (TN+TP)/(TN+FP+FN+TP)

- 정밀도와 재현율

둘 다 positive 데이터 세트의 예측 성능에 초점을 맞춘 평가 지표

- 정밀도

= TP / (FP+TP)

예측을 positive한 대상 중에서 예측과 실제 값이 positive로 일치한 데이터의 비율

양성 예측도라고도 불림

실제 negative인 데이터 예측을 positive로 잘못 판단하게 되면 업무상 큰 영향이 발생하는 경우 중요한 지표가 된다. (ex. 스팸 메일 여부 판단 모델)

- 재현율

= TP/(FN+TP)

실제 값이 positive한 대상 중에 예측과 실제 값이 positive로 일치한 데이터의 비율

민감도 또는 TPR이라고도 불림

실제 positive 데이터를 negative로 잘못 판단하게 되면 업무상 큰 영향이 발생하는 경우 재현율이 중요한 지표가 된다. (ex. 암 판단 모델)

- 정밀도/재현율 트레이드오프

정밀도 또는 재현율이 특별히 강조돼야 할 경우 분류의 결정 임곗값을 조정해 정밀도 또는 재현율의 수치를 높일 수 있음.

하지만 둘은 상호 보완적인 평가 지표이기 때문에 한 쪽을 강제로 높이면 다른 하나의 수치는 떨어질 수밖에 없다.

분류 결정 임곗값이 낮아질수록 positive로 예측할 확률이 높아짐 -> 재현율 증가

- F1 스코어

정밀도와 재현율을 결합한 지표

정밀도와 재현율이 어느 한쪽으로 치우치지 않는 수치를 나타낼 때 높은 값이 나옴

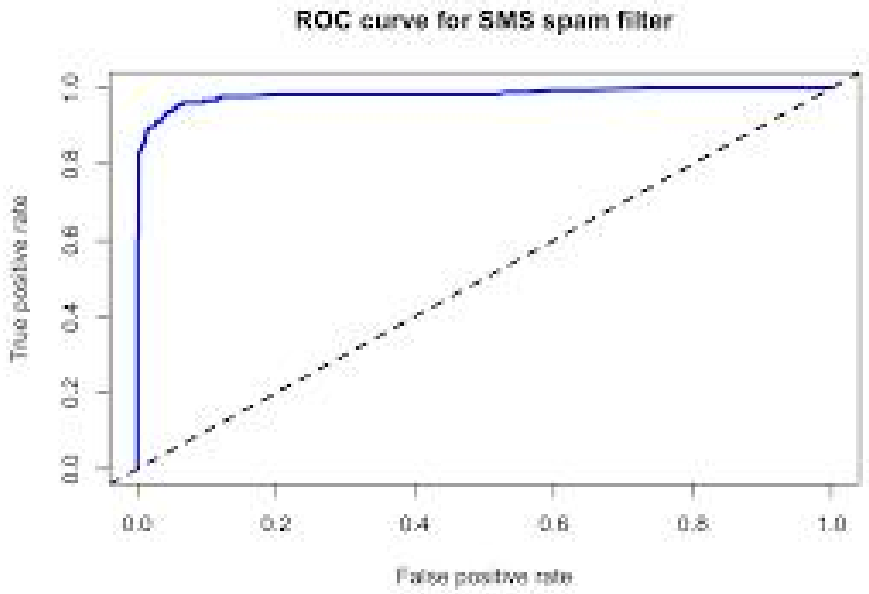
$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 * (precision * recall)}{precision + recall}$$

- ROC 곡선과 AUC

이진 분류의 예측 성능 측정에서 중요하게 사용되는 지표

- ROC 곡선

  수신자 판단 곡선으로 FPR (false positive rate)이 변할 때 TPR (true positive rate. 재현율)이 어떻게 변하는지를 나타냄



  가운데 직선이 roc 곡선의 최저값이고 이에 가까울수록 성능이 떨어지는 것