Week 14

데이터기반 분류 예측

데이터 소개 & 주의사항

- Red Wine Quality Dataset
  - 각 와인 별 주요 특징들을 종합하여, 'quality'가 매겨집니다.
  - Output variable: Quality
  - Input Variable: Fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol

- Dataset 출처 및 주의사항
  - 출처: Red Wine Quality | Kaggle
  - 구름에서 제공해 드리는 데이터셋을 활용하세요.

- 학습 시 주의사항
  - 학습 시 warning이 발생한다면 max_iter 값의 문제일 수 있습니다.
  - 조교의 경우 1000으로 했을 때 문제가 되지 않았으니, 진행에 참고하세요.

- 정답 우회 시
  - 문제를 올바르게 풀지 않고 출력만 조작하거나, dataset 분배를 임의로 조작하지 마세요.
  - 제출 코드에 해당 내용이 발견될 경우 불이익이 발생할 수 있습니다.

데이터 소개

항목	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	Quality (Output)
해설	고정 산도	휘발 산도	구연산	잔여당	염화물	와인 등급
값	4.6~15.9	0.12 ~ 1.58	0~1	0.9 ~ 15.5	0.01 ~ 0.61	3~8
항목	**free sulfur dioxide**	**total sulfur dioxide**	**density**	**pH**	**sulphates**	
해설	이산화황	이산화황	밀도	pH	황산염	
값	1~72	6~289	0.99~1	2.74~4.01	0.33~2	

1 Quality Class Quantization

- 여러분은 와인이 추천할만한 와인인지, 아닌지를 판정하는 모델을 만들고자 합니다.
  - 아래의 조건에 따라 quality 열의 값을 대체하세요.
  - Quality가 3~5인 경우: '0'으로 변경
  - Quality가 6~8인 경우: '1로 변경
  - 변경 불가능한 하단 코드에서 0, 1로 분류된 와인 수를 출력하는 것을 통해 채점합니다.

Quality
7
3
4
8
6

Quality
1
0
0
1
1

```
1  import pandas as pd
2
3  df = pd.read_csv('data/winequality-red.csv',dtype=float)
4
5
6  print(len(df.loc[df['quality'] == 0]))
7  print(len(df.loc[df['quality'] == 1]))
```

2 Binary Classification 모델 학습

- 학습 규칙
  - 전체 데이터 중 [0:300] 데이터는 testset으로 활용됩니다.
  - 전체 데이터 중 [300:] 데이터를 trainset으로 활용됩니다.
  - 코드 상단부에 잠금 형태로 지정해 두었기에, 세트 자체는 변하지 않도록 주의하세요.
  - Quality를 제외한 10개의 독립 변수를 사용하는 분류 모델을 설계하세요.
  - 주어진 데이터는 0, 1의 분류가 포함되고, 원래의 3~8 범위의 quality는 빠져 있습니다.

- 모델 목표
  - 학습 세트에 대한 정답률 0.7 이상
  - 테스트 세트에 대한 정답률 0.7 이상
  - Classification 성공률이 달성될 경우 정답으로 간주됩니다.
  - 둘 모두에 달성할 경우 “True”가 총 2회 출력됩니다.

```
8  train_score = lr.score(x, y)
9  test_score = lr.score(test_x,test_y)
10 train_pass = False
11 test_pass = False
12
13 if(train_score >= 0.7):
14   train_pass = True
15
16
17 if(test_score >= 0.7):
18   test_pass = True
19 print(train_pass,test_pass)
```

3 Binary Classification 모델 학습

- 데이터 탐색
  - 데이터 탐색을 통해 모델의 성능을 높이는 것이 여러분의 목표입니다.
  - **상관계수** 및 **시각화**를 통한 데이터 탐색을 거쳐, 모델에 활용할 독립 변수를 자유롭게 선정하세요.
  - Hint: 이상치 제거의 경우 total/free sulfur dioxide를 살펴보세요.
- 학습 규칙
  - 학습 규칙은 2번 문제와 동일합니다.
  - 독립변수 수는 데이터 탐색을 통해 자유롭게 활용하세요.
- 모델 목표
  - 학습 세트에 대한 정답률 0.735 이상
  - 테스트 세트에 대한 정답률 0.715 이상
  - Classification 성공률이 달성될 경우 정답으로 간주됩니다.
  - 둘 모두에 달성할 경우 “True”가 총 2회 출력됩니다.

```
train_score = lr.score(x, y)
test_score = lr.score(test_x,test_y)
train_pass = False
test_pass = False

if(train_score >= 0.735):
  train_pass = True


if(test_score >= 0.715):
  test_pass = True
print(train_pass,test_pass)
```