Data Report

5조 김나경 김수민 김정하 양승준 이유정

0. github 주소

https://github.com/urcloud/Wine_Quality.git

1. Data Set 개요

출처: Kaggle "Red Wine Quality" data set (총 1599개 데이터셋)
https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009/data

입력 변수
- fixed acidity (고정 산도)
- volatile acidity (휘발성 산도)
- citric acidity (구연산 함량)
- residual sugar (잔류 당분)
- chlorides (염화물 함량)
- free sulfur dioxide (유리 이산화황)
- total sulfur dioxide (총 이산화황)
- density (밀도)
- pH (산도)
- sulphates (황산염 함량)
- alcohol (알코올 도수)

target 값
quality(품질)

2. Data Cleaning

2.1 결측치 확인

```
nan_mask = np.isnan(data)

# 결측값 포함된 행 번호
nan_rows = np.where(np.isnan(data).any(axis=1))[0]

# (row, col) 위치
nan_positions = np.argwhere(np.isnan(data))

print("===== 결측값이 있는 행 번호 =====")
print(nan_rows)

print("\n===== 결측값 위치 (row, column) =====")
for r, c in nan_positions:
    print(f"Row {r}, Column {c} ({columns[c]})")
```

```
===== 결측값이 있는 행 번호 =====
[]

===== 결측값 위치 (row, column) =====
```

결측치 없음을 확인했습니다.

2.2 이상치 확인 (관계 기반 이상치)

정상 데이터들을 압축했다가 다시 복원하는 Autoencoder를 학습시키고, 각 샘플의 재구성 오차가 다른 샘플들에 비해 크면 이상치로 판단했습니다. 임계값은 재구성 오차의 평균 + 3 * 표준편차로 정했습니다.

행 번호	검출 횟수 (총 40회)	이상치 판별
13	19	
17	11	
33	18	
81	36	0
83	15	
86	40	0
91	40	0
92	40	0
106	38	0
142	1	
144	1	
151	40	0
169	39	0
226	26	0
258	39	0
281	13	
324	30	0
325	30	0
353	1	
354	11	
396	19	
400	19	
415	1	
442	1	
451	19	
467	2	
480	40	0
544	1	
554	1	
555	1	
557	2	
559	2	
564	2	

행 번호	검출 횟수 (총 40회)	이상치 판별
584	2	
591	1	
614	10	
651	1	
652	37	0
672	9	
692	14	
723	39	0
754	28	0
861	35	0
1017	33	0
1018	33	0
1051	31	0
1079	38	0
1081	39	0
1114	10	
1131	12	
1165	1	
1235	40	0
1244	37	0
1260	15	
1269	2	
1270	1	
1299	18	
1319	34	0
1370	29	0
1372	29	0
1434	38	0
1435	38	0
1474	35	0
1476	35	0
1558	6	
1574	40	0

threshold를 0.5로 잡아 총 40번의 결과 중 20번 이상 검출된 값을 이상치로 판별했습니다.
이상치는 총 31개로, 각 이상치의 행 번호는 다음과 같습니다.
[81, 86, 91, 92, 106, 151, 169, 226, 258, 324, 325, 480, 652, 723, 754, 861, 1017, 1018, 1051, 1079, 1081, 1235, 1244, 1319, 1370, 1372, 1434, 1435, 1474, 1476, 1574]

2.3 이상치 처리 방식

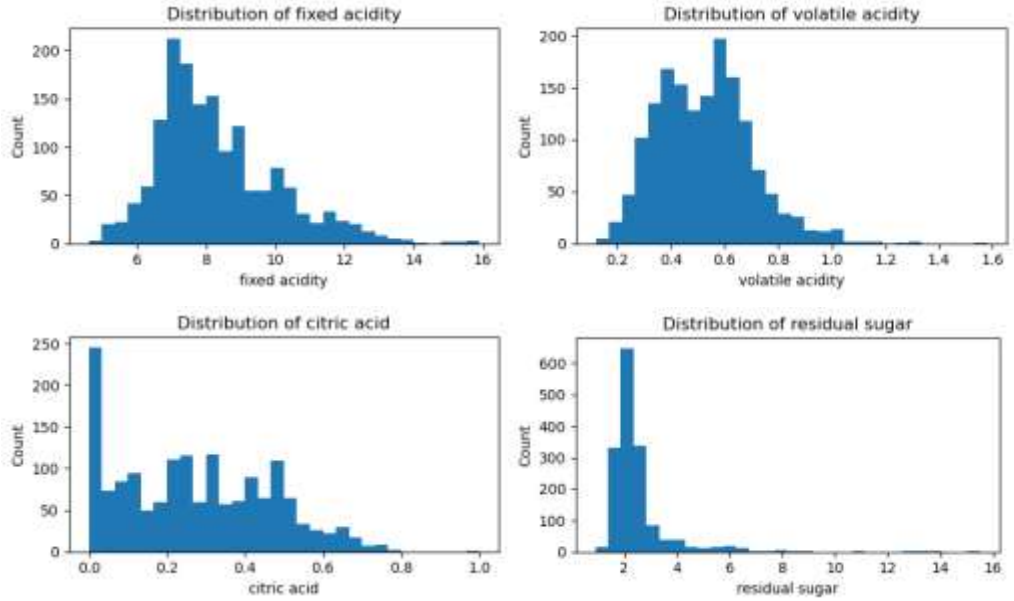
전체 데이터 1,599개 중 약 31개(약 1.9%)가 이상치로 확인되었으며, 비율이 높지 않아 제거하지 않고 그대로 활용하기로 했습니다. 다만 이러한 이상치가 모델 학습에 불필요한 영향을 줄 수 있어, MSE 대비 이상치에 둔감한 Huber Loss를 손실 함수로 사용함으로써 안정적인 DNN 모델을 개발하고자 합니다.
Huber Loss는 작은 오차 구간에서는 MSE처럼 작동하고, 큰 오차 구간에서는 MAE처럼 작동하여 이상치에 크게 영향받지 않고 안정적인 모델 학습이 가능한 손실 함수입니다.

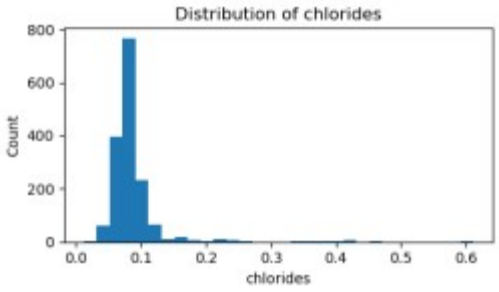
3. Feature summary

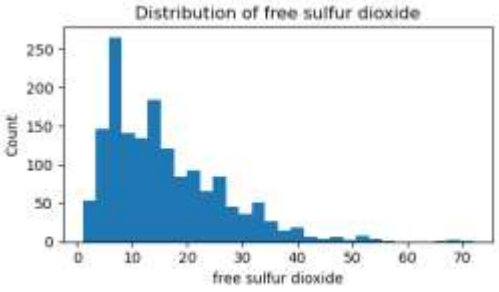
3.1 Feature summary

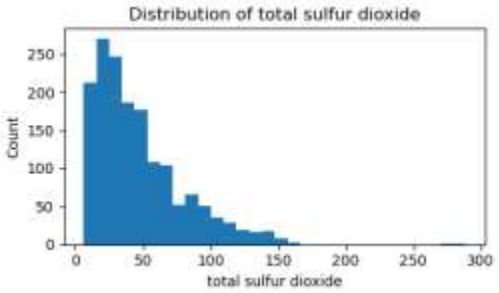
	MIN	MAX	MEAN	STD
fixed acidity	4.6	15.9	8.3196	1.7406
volatile acidity	0.12	1.58	0.5278	0.1790
citric acid	0	1	0.2710	0.1947
residual sugar	0.9	15.5	2.5388	1.4095
chlorides	0.012	0.611	0.0875	0.0471
ree sulfur dioxide	1	72	15.8749	10.4569
total sulfur dioxide	6	289	46.4678	32.8850
density	0.9901	1.0037	0.9967	0.0019
pH	2.74	4.01	3.3111	0.1543
sulphates	0.33	2	0.6581	0.1695
alcohol	8.4	14.9	10.4230	1.0653
quality	3	8	5.6360	0.8073

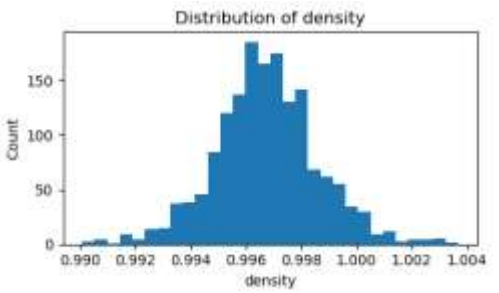
3.2 각 특징 별 분포 시각화

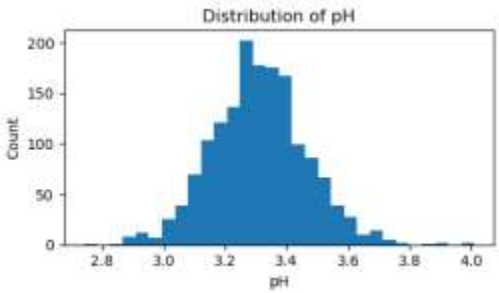


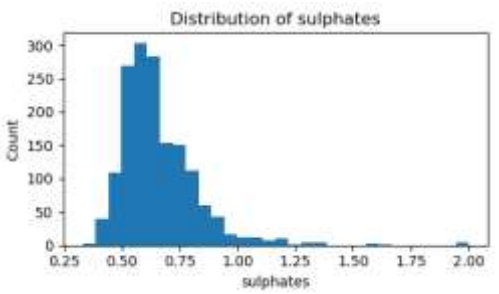

Distribution of chlorides
Count
chlorides

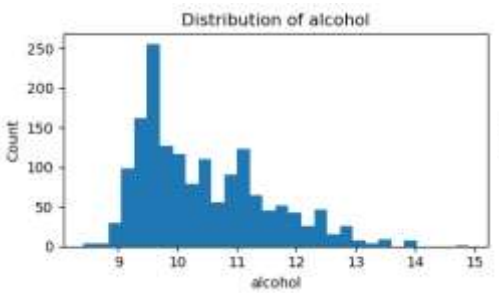

Distribution of free sulfur dioxide
Count
free sulfur dioxide

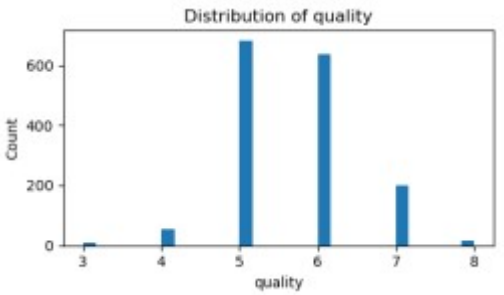

Distribution of total sulfur dioxide
Count
total sulfur dioxide


Distribution of density
Count
density


Distribution of pH
Count
pH


Distribution of sulphates
Count
sulphates


Distribution of alcohol
Count
alcohol


Distribution of quality
Count
quality

3.3 정규화 방식

딥러닝 모델에서는 입력 변수들의 스케일이 크게 다를 경우 학습 과정이 불안정해지고 성능이 저하될 수 있습니다. 스케일이 큰 변수가 스케일이 작은 변수에 비해 손실 함수 계산 시 더 큰 영향을 미치게 되어 모델이 해당 변수 중심으로만 학습하는 편향이 발생할 수 있습니다. 딥러닝 모델은 가중치 업데이트를 Gradient Descent에 기반하여 수행하는데, 입력 스케일이 클 경우 gradient의 크기도 함께 커져 학습이 과도하게 요동치거나 발산할 위험이 있습니다. 반대로 입력 스케일이 작은 변수는 기울기가 매우 작아져 학습이 거의 이루어지지 않는 vanishing gradient 문제가 나타날 수 있습니다. 이처럼 변수 간 단위 차이가 크게 나면 전체 학습 속도가 느려지고, 최적해에 수렴하기 어려우며, 결국 모델의 예측 성능이 떨어질 가능성이 높아집니다.

모든 특성을 유사한 스케일로 맞추는 과정인 정규화를 적용하면 모델이 변수의 단위 차이에 영향을 받지 않고 데이터 본연의 패턴을 안정적으로 학습할 수 있으며, 기울기 계산 또한 안정화되어 학습 속도가 향상되고 수렴이 보다 원활해집니다. 각 데이터의 스케일을 시각화 한 결과, 각 데이터의 스케일이 달라 정규화 과정이 필요하다고 판단해 적용할 정규화 방법에 대해 논의했습니다.

Z-score 정규화는 각 변수에서 평균을 제거하고 표준편차로 나누어 변수를 표준 정규분포 형태로 변환하는 방식입니다. 이 방법은 데이터 내 이상치가 존재하더라도 Min-Max 정규화에 비해 극단값의 영향을 덜 받는다는 장점이 있습니다. Min-Max 정규화는 변수의 최소값과 최대값에 직접적으로 의존하기 때문에 이상치가 포함될 경우 전체 데이터가 좁은 구간으로 압축되는 문제가 발생합니다. 반면 Z-score는 평균 중심의 정규화이기 때문에 이상치가 일부 존재하더라도 전체 스케일 변화가 크게 왜곡되지 않습니다. 따라서 본 데이터셋에서는 전체 1,599개 중 약 1.9% 수준의 이상치가 확인되었으므로, 이상치를 제거하지 않고 학습에 포함하되 모델 성능이 왜곡되지 않도록 보다 안정적인 정규화 방식인 Z-score를 적용하기로 했습니다.

4. 팀 미팅

줌으로 팀 미팅 진행했습니다.