먼저 데이터 처리, 시각화 및 기계 학습을 위한 몇 가지 일반적인 라이브러리를 도입했습니다.load\_and\_preprocess\_data 함수를 대입하여 CSV 파일에서 데이터를 로드하고 누락된 값을 처리하고 데이터를 특성(X)과 표적 변수(y)로 나눈 다음 특성을 표준화합니다.그런 다음 EDA를 사용하여 데이터의 기술 통계를 인쇄하고 상자 그림 및 히스토그램을 그리고 특성 간의 열역학을 그려 상관 관계를 보여주었습니다.그런 다음 데이터는 split\_data 함수를 사용하여 훈련 세트와 테스트 세트로 나누었으며, 그 중 80%는 훈련에, 20%는 테스트에 사용되었습니다.그런 다음 데이터를 대입하고 데이터 열 'quality'의 고유한 값을 인쇄합니다.

다음으로 나는 PyTorch를 사용하여 와인의 품질 범주를 예측하는 데 사용되는 신경망 모델을 구성, 훈련 및 평가했습니다.첫 번째는 신경망이 일반적으로 출력을 정수로 예상하기 때문에 map\_quality 함수를 통해 품질을 정수로 매핑하는 데이터 전처리입니다.여기에서 품질 범주는 3에서 8까지이며 0에서 5까지의 정수로 매핑됩니다.데이터를 PyTorch에서 처리하고 훈련할 수 있도록 PyTorch 텐서로 변환했습니다.대규모 데이터 세트의 훈련에 필요한 배치로 데이터를 로드할 수 있도록 WineDataset 클래스 및 데이터 로더를 생성합니다.배치 훈련은 모델의 수렴과 메모리 사용 감소에 도움이 됩니다.다음으로 신경망 모델(WineQualityNet)을 정의하며, 이 모델에는 여러 개의 전체 연결 계층과 dropout 계층이 포함됩니다.이러한 구조의 선택은 일반적으로 경험과 작업 요구 사항을 기반으로 합니다.그런 다음 손실 함수와 최적화 기능이 있는데, 이는 다중 범주 분류 문제에 대한 일반적인 선택인 교차 엔트로피 손실 함수(Cross Entropy Loss)를 사용합니다.여기에서 저는 Adam 최적화 장치를 선택했습니다.모델 훈련은 훈련 데이터의 완전한 통과를 포함하는 여러 기간(epochs)의 반복을 포함하여 train\_model 함수에 의해 수행되며 모델의 무게는 손실에 따라 업데이트됩니다.마지막으로 evaluate\_model 함수로 모델을 평가하고 테스트 데이터 세트를 사용하여 모델의 정확도를 계산합니다.최종 모델의 평균 성능은 0.6245로 이전 모델의 평균 정확도 0.596875보다 높으며 개선되었다고 볼 수 있습니다.