

MODEL REPORT

1. Preprocessing

- Data Transformation

원본 .hea 파일을 읽어와서, II 리드에 해당하는 신호를 추출하였음. 이 과정에서 wfdb 라이브러리를 사용하였으며, 추출된 신호는 numpy 배열로 변환하여 저장하였음.

- Resampling

ECG 신호의 일관성을 유지하도록 신호의 길이와 샘플 레이트를 조정하였음. 이 과정에서 scipy 라이브러리를 활용하였으며, 10 초 길이의 신호를 250Hz 샘플 레이트로 재샘플링하여 총 2500 개의 샘플을 갖도록 하였음.

- Standardization

각 신호의 mean 과 standard deviation 를 계산한 후, 신호에서 mean 을 빼고 standard deviation 로 나누어 표준화를 수행하였음. 이 과정에서 작은 값의 표준 편차를 피하기 위해 작은 epsilon 값을 추가하였음.

- Filtering

신호의 noise 를 제거하고 분석에 필요한 특정 주파수만 통과시키기 위해 scipy 라이브러리의 butter 와 filtfilt 함수를 사용하여 필터를 설계하고 적용하였음. 주요한 정보들을 대부분 담고 있는 0.5Hz 에서 50Hz 사이의 주파수 필터링을 시행하였음.

- TensorDataset Undersampling

전처리된 신호 데이터를 TensorDataset 으로 변환하고, DataLoader 를 사용하여 모델 학습에 필요한 배치 단위로 데이터를 로드하였음. Training data(p1000-p1021)의 샘플들은 총 16734 세트로 이루어져 있었으나, 1 과(Atrial_Fibrillation 과 Atrial_Flutter) 0(etc)의 샘플 개수 차이가 0 이 1 의 10 배 정도 많았기 때문에 언더샘플링을 수행함. 각 1806 개, 1806 개로 총 3612 개 세트의 샘플로 모델 훈련 시행하였음.

- Data Augument

모델의 일반화 능력을 향상시키기 위해 다양한 변환을 적용하여 데이터를 증가시켰음. 신호의 길이를 늘리거나 줄여서 시간 축에서 변형을 주는 Time Stretching, 랜덤으로 노이즈를 추가하는 것, 신호를 시간 축에서 좌우로 이동시켜 위치 변화를 주는 방법 등을 수행함.

2. Modeling & Training

가장 기본적인 합성곱 신경망(1D CNN) 아키텍처를 적용함. 모델은 3 개의 convolutional layer 와 batch normalization, ReLU activation function, dropout layer, full connect layer 로 구성되었음. 시계열 형태의 신호 데이터의 feature 를 효과적으로 추출할 수 있도록 설계함.

- 하이퍼파라미터

- 학습률(Learning Rate): 0.0001
- 배치 크기(Batch Size): 16

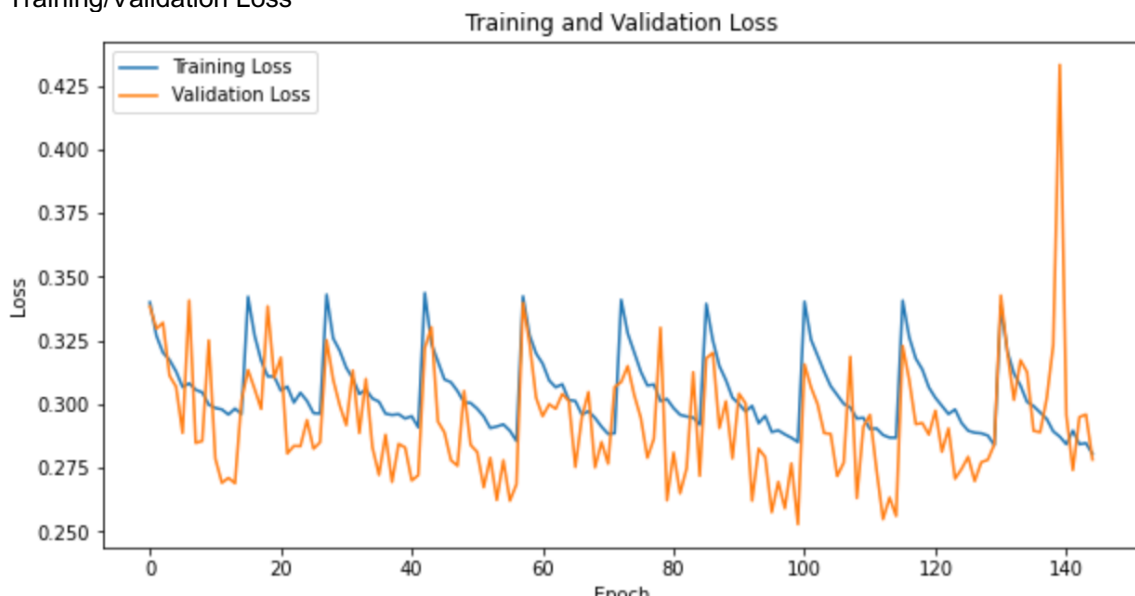
- 에포크 수(Number of Epochs): 15
- 가중치 초기화(Weight Initialization): Xavier 초기화
- 손실 함수(Loss Function): 이진 교차 엔트로피 손실 (BCEWithLogitsLoss)
- 최적화 알고리즘(Optimizer): Adam 옵티마이저, L2 정규화 적용 (Weight Decay)

· Training and Validation

- 모델 학습은 교차 검증(Cross-Validation) 기법을 통해 이루어짐. Overfitting 방지를 위해서 early stopping 방식을 적용함. 또한 학습 중의 훈련 손실과 검증 손실을 그래프로 시각화하고, Confusion Matrix 를 통해 모델의 성능을 평가함.
- 또, ROC 곡선과 Youden's J 통계를 사용하여 최적의 분류 임계값을 찾음. 애초에 validation 과 evaluation 데이터셋들은 언더샘플링을 진행하지 않아, binary classification 에서의 데이터 불균형이 상당히 심함. 그대로 시그모이드 함수로 확률을 구해버리면 0 값을 기준으로 normal distribution 을 따르기 때문에 cutoff 를 0.1 수준으로 조정함. (실제로 Youden's J 통계를 통해 확인)

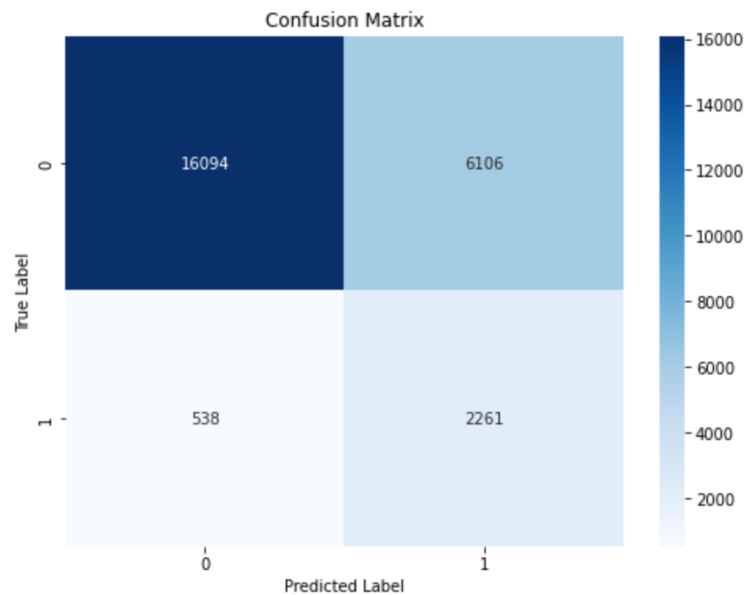
3. Evaluation

· Training/Validation Loss



- 학습 초반에 빠르게 감소한 후, 에폭이 진행될수록 비교적 안정된 패턴을 보이는 것을 확인. 전반적으로는 매 fold 단계마다 15 epochs 씩 진행될 수록 training loss 보다 validation loss 가 적은 값을 보여 오버피팅이 없음을 확인함.
- 다만 마지막 10 번째 fold 에서 중간 에폭 지점에서 Validation Loss 가 확 증가하는 현상이 보였는데, 이는 해당 fold 의 데이터들에 한해 모델이 그와 유사한 feature 의 일부 데이터에 대해 과적합(overfitting)될 가능성을 확인. 이렇게 학습 도중 발생하는 일부 데이터셋에서의 Validation Loss 의 상승은 데이터의 복잡성과 불균형성에 기인함.
-> 그래서 차후 K-Fold Cross Validation 을 통해 모델의 안정성을 평가하고 다양한 fold 에 대한 평균 성능을 계산하여 신뢰도를 높일 필요가 있을 것으로 보임.

· Confusion Matrix



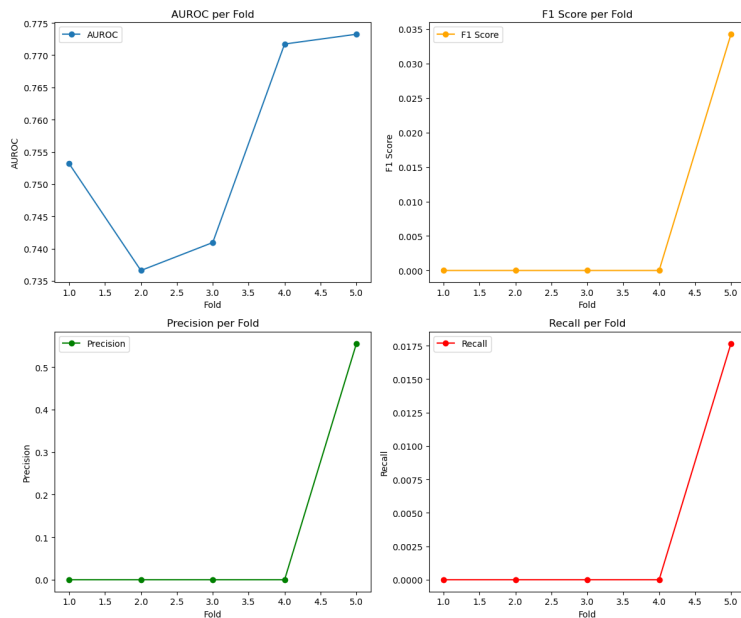
- True Positive: 2261
- True Negative: 16094
- False Positive: 6106
- False Negative: 538
- Negative 영역에서 크게 문제가 되고 있는 사항, 즉 over detection 이나 under detection 은 없음을 확인. 다만 문제는 positive 영역에서의 over detection 이 상당히 보인다는 점.
- 모델의 강점: 높은 True Positive 와 True Negative 비율을 통해 대부분의 예측을 정확히 수행하는 것을 확인할 수 있었음.
- 모델의 개선 필요점: False Positive 와 False Negative 비율이 상대적으로 높아, 특히 부정 클래스 예측 시 과잉 탐지 문제를 해결하기 위해 추가적인 조치가 필요함.

	precision	recall	f1-score	support
Class 0	0.96	0.80	0.87	3655
Class 1	0.35	0.78	0.48	495
accuracy			0.80	4150
macro avg	0.65	0.79	0.68	4150
weighted avg	0.89	0.80	0.83	4150

- 모델의 전체 정확도는 80%로 양호하지만, 역시나 Class 1 의 TP와 FP를 활용해 계산하는 precision 값은 확연히 떨어짐. 그에 따른 F1 score 도 class 0 에 비해 현저히 떨어짐을 확인.
- 모델은 Class 0 (Negative Class)에 대해 매우 높은 Precision 과 상대적으로 높은 Recall 을 보임. 이는 대부분의 negative 샘플을 정확히 예측하고 있으며, 일부 positive 샘플을 놓치고 있다는 것을 의미
- 모델의 Class 1 (Positive Class)에 대한 Precision 이 매우 낮음. 이는 positive 로 예측한 샘플 중 많은 부분이 실제로는 negative 임을 나타냄. Recall 은 비교적 높지만 Precision 이 낮아 전체 F1-score 가 낮음.

- Class 1 에 대한 Precision 을 높이기 위해 SMOTE 등의 방법을 사용하거나, cost-sensitive learning 을 적용하는 방법을 고려해야할 듯함.

참조



- 데이터의 어떤 전처리도 없이(리샘플링, 언더샘플링, 표준화 등등) 진행했을 때의 결과값을 참고해서 보면 대부분 AUROC, F1 Score, Precision, Recall 에서 모두 낮은 점수를 기록하고 있음. 이는 데이터의 잡음과 불균형이 그대로 반영되었기 때문. 전처리 단계에서의 개선을 통해 모델의 성능을 향상시키는 방향으로 진행할 필요가 충분히 있음.