# ADS5006 기계학습특론
# 기말과제 부정맥 판별


| 과목 | 기계학습특론 |
|---|---|
| 이름 | 김한비 |
| 소속 | 데이터사이언스융합과 |
| 학번 | 2021712203 |

# 목차

# 1. 연구 주제 소개

## 1.1 데이터 소개

이번 과제에 사용된 데이터는 VT/VF 가 없는 환자 데이터 619 개, VT/VF 가 있는 환자데이터 27 개, 그리고 테스트를 위한 285 개의 데이터가 존재한다. 각각의 환자 데이터는 크게 두가지로 분류 할 수있다. 하나는 ECG 를 통해 얻어낸 전체적인 ECG Value 값이며 이는 아래 보이는 표 (Fig 1)로 요약 할 수있다. 두 번째, 데이터는 12 ECG Leads 데이터로 각 환자 별로 총 12 개의 ECG Leads wave 에 대해, 하나의 wave 당 5000 개의 데이터가 존재한다. 아래 표(Fig 2)를 통해 VT/VF 가 있는 환자와 없는 환자의 12 개의 Lead 의 차이를 볼 수 있다.

| 변수명 | ECG Value Data 설명 |
|---|---|
| patientID | 환자 데이터에 할당된 아이디 |
| Gender | 성별 ('Male', 'Female') |
| Age | 나이 |
| HeartRate | 심장박동수 |
| PRInterval | 심방 분극과 심실 분극 한 셋트 사이의 시간 |
| RRInterval | 심실 박동율 |
| QRSDuration | 심실근 분극 시간 |
| QTInterval | 심실 분극 후 재분극 까지의 시간 |
| QTCorrected | 심장 박동 60bpm을 기준으로 추정된 QTInterval 값 |
| PAxis | ECG에서 측정된 P-Axis 값 |
| RAxis | ECG에서 측정된 R-Axis 값 |
| TAxis | ECG에서 측정된 T-Axis 값 |
| labels | VT/VF 판별 유무, (0.0=No Symptom, 1.0 = VT/VF) |

Fig 1. ECG Value 데이터 소개 및 변수 설명 ( 참조 https://ecg.utah.edu/lesson/1)

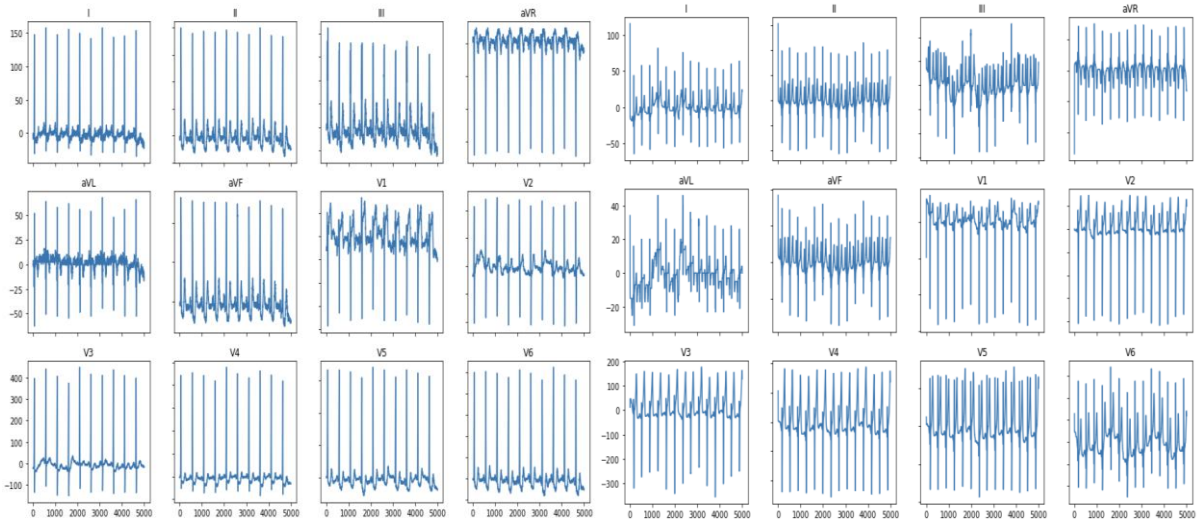


Fig 2. VT/VF가 없는 환자 (왼쪽), VT/VF가 있는 환자(오른쪽) 12 ECG leads 그래프 비교

### 1.2 문제정의

The standard of 12 lead ECG(Electrocardigram) 데이터를 활용하여 VF(Ventricular fibrillation) 와 VT(pulseless ventricular tachycardia)를 판별하는 머신러닝 알고리즘 비교 판별한다. 본 연구의 목적은 VT/VF 가 있는 환자와 없는 환자를 구분하는 Binary Classification 문제이다. 주어진 데이터가 불균형하고 데이터의 종류가 많음으로 두 가지 방법을 제시하고 이를 비교하려 한다. 따라서, 불균형 데이터를 처리하는 전처리 과정이 추가적으로 필요하다. 첫번쨰로, ECG Value Data 를 머신러닝 알고리즘을 통해 훈련시키고 해당하는 손실 함수와 정확도를 비교한다. Binary Classification 에 특화된 Logistic regression 과 전체적으로 성능이 좋은 RandomForestClassification 을 비교해보고 첫번째 방법의 주요 알고리즘으로 선택한다. 두번쨰로, 12 ECG Leads Data 를 통해 또 다른 머신러닝 알고리즘 기법을 적용하고 앞선 머신러닝 기법과 비교한다. 두번째 방법에서는, 앞서 소개한 대로 12 ECG Lead 데이터가 각 환자 당 5,000 개씩 12 개가 존재함으로 데이터의 크기가 훨씬 크다. 따라서 PCA 차원 축소를 진행하고 또한 Neutral Network 기반의 알고리즘을 통해 최종 알고리즘을 선정한다. 최종적으로 두 가지 다른 데이터와 머신러닝 알고리즘을 비교하여 가장 좋은 성능을 내는 방법을 정의하고 이를 Test 데이터에 적용한다.

## 2. 머신 러닝 방법 I

### 2.1 데이터 전처리

ECG Value Data 의 경우, 다수의 결측치가 존재하였다. 데이터의 수가 비교적 많지 않고 VF/VT 환자 데이터가 압도적으로 부족하기 때문에, 제거하여 기존 데이터에 대한 손실을 가져오기 보다, 결측치를 새로운 값으로 대체하는 것이 더 효율적이라고 판단된다. Null 값이 존재하는 변수는 총 네개로 'PRInterval', 'RRInterval', 'PAxis', 'TAxis' 가 그에 해당한다. 'RRInterval'의 경우 상관계수 분석 (Fig 3) 을 통해 'HeartRate'와 'QTInterval'과 높은 상관 관계를 보이는 것이 확인 되었고, 차후 분석에서 제거하였다. 나머지 세 변수에 대해서는 평균으로 대체하는 방법과 각각의 변수를 머신러닝 알고리즘을 통해 예측하는 방법을 비교하여 RMSE 가 낮은 방법을 채택하였다. 머신머닝 모델 RandomForestRegressor 를 통해 각각의 변수를 예측하여 사용하는 것이 평균을 사용하는 것보다 RMSE 가 더 낮았기 때문에, 해당 변수들의 결측치는 나머지 변수들에 의해서 예측된 값으로 대체하였다.

| | Age | HeartRate | PRInterval | RRInterval | QRSDuration | QTInterval | QTCorrected | PAxis | RAxis | TAxis | labels |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Age | 1.000000 | -0.022472 | 0.118032 | 0.055326 | 0.040896 | 0.119865 | 0.118729 | -0.050361 | -0.147083 | 0.066201 | -0.082939 |
| HeartRate | -0.022472 | 1.000000 | -0.055401 | -0.950868 | -0.142119 | -0.732370 | 0.240233 | 0.076713 | -0.094085 | 0.033540 | -0.151526 |
| PRInterval | 0.118032 | -0.055401 | 1.000000 | 0.096187 | 0.143072 | 0.104351 | 0.045874 | -0.095006 | -0.053222 | 0.013128 | 0.170010 |
| RRInterval | 0.055326 | -0.950868 | 0.096187 | 1.000000 | 0.127914 | 0.761865 | -0.212373 | -0.105383 | 0.100282 | -0.050146 | 0.166491 |
| QRSDuration | 0.040896 | -0.142119 | 0.143072 | 0.127914 | 1.000000 | 0.263608 | 0.222171 | -0.000875 | -0.110111 | -0.037835 | 0.117095 |
| QTInterval | 0.119865 | -0.732370 | 0.104351 | 0.761865 | 0.263608 | 1.000000 | 0.459580 | -0.100722 | 0.025341 | 0.011749 | 0.207803 |
| QTCorrected | 0.118729 | 0.240233 | 0.045874 | -0.212373 | 0.222171 | 0.459580 | 1.000000 | -0.022266 | -0.097184 | 0.067666 | 0.084989 |
| PAxis | -0.050361 | 0.076713 | -0.095006 | -0.105383 | -0.000875 | -0.100722 | -0.022266 | 1.000000 | 0.106982 | 0.015522 | 0.017918 |
| RAxis | -0.147083 | -0.094085 | -0.053222 | 0.100282 | -0.110111 | 0.025341 | -0.097184 | 0.106982 | 1.000000 | -0.045171 | 0.003591 |
| TAxis | 0.066201 | 0.033540 | 0.013128 | -0.050146 | -0.037835 | 0.011749 | 0.067666 | 0.015522 | -0.045171 | 1.000000 | -0.105839 |
| labels | -0.082939 | -0.151526 | 0.170010 | 0.166491 | 0.117095 | 0.207803 | 0.084989 | 0.017918 | 0.003591 | -0.105839 | 1.000000 |

Fig 3. 상관 관계 분석표

최종적으로 SMOTE 기법을 활용하여, VF/VT가 있는 환자의 데이터를 없는 환자의 데이터 수와 동일하게 맞춰줌으로써 불균형 데이터에 대한 문제를 Minority Class에 대해 Oversampling 하는 것으로 진행하였다. 이 경우, Majority Class에 대해 Undersampling을 진행 할 수도 있지만, 개수가 27개 임으로 데이터의 손실이 너무 커지는 것을 우려하여 SMOTE을 통한 Oversampling이 더 적절하다고 판단하였다.

## 2.2 모델링: Logisitic Regression & Random Forest

앞서 전처리가 완료한 데이터는 각각 0.3의 비율로 훈련/테스트 데이터셋으로 구분한 후, Logistic regression과 Random Forest 머신러닝 알고리즘을 적용하였다. 데이터의 불균형함이 SMOTE을 통해 어느정도 해결되었기 때문에, Accuracy을 해당 모델을 비교하고 판단하는데 사용하기로 결정하였다.

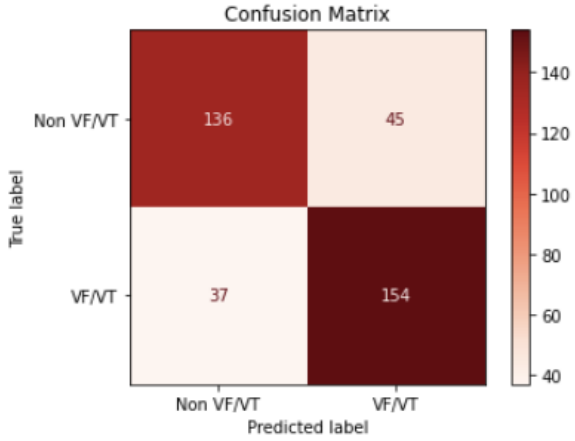


Fig 4. Confusion Matrix for Logistic Regression

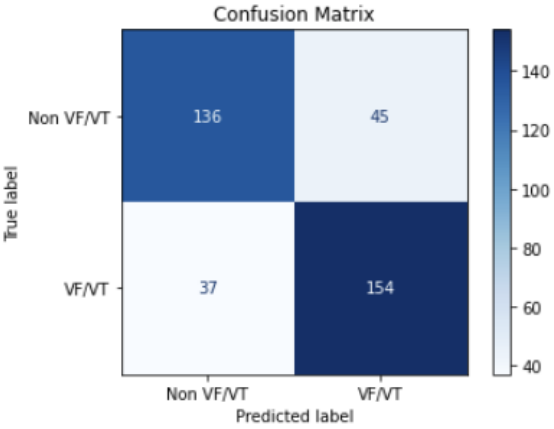


Fig 5. Confusion Matirix for Random Forest

위에 보이는 Confusion Matrix를 통해, Random Forest의 결과가 Accuracy 측정 면에서 0.91으로 Logistic regression의 0.77에 비해 상당히 높았다. 하지만, Random Forst의 특성상 Accuracy가 높았다는 것은 현재 데이터에 대해 Overfitting이 될수 있음을 의미하기도 한다. 따라서, Feature Importance를 측정하여 현재 사용된 변수의 중요도를 판단하고 중요도가 너무 낮은 변수를 제거한 후 Cross-Validation을 통해 Random Forest의 Overfitting 가능성을 낮추는 시도를 하였다.

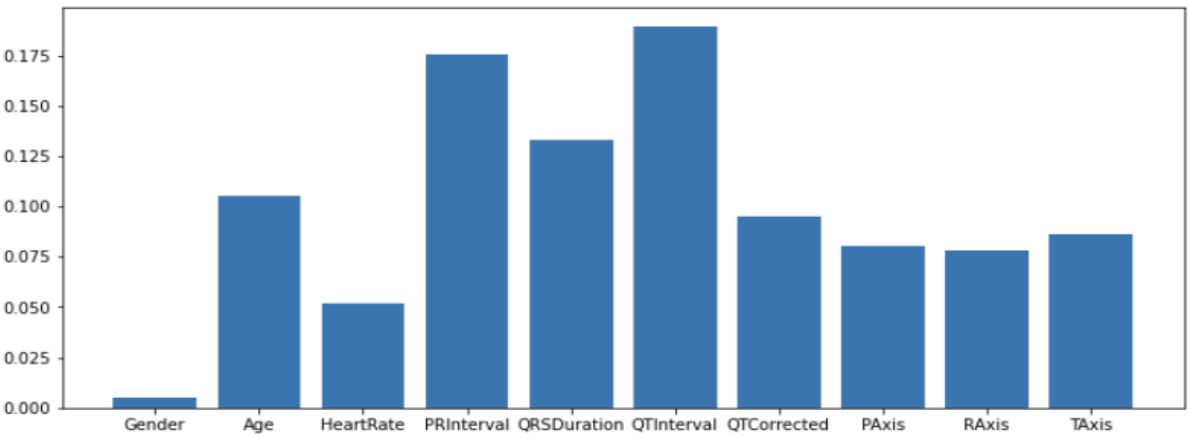


Fig 6. Random Forest에 대한 각 변수 별 Feature Importance

Feature Importance 측정결과, 'Gender' 변수의 경우 결과를 예측하는데 다른 변수들의 비해 중요도가 현저히 낮았다. 이는 VF/VT는 성별에 많은 영향을 받지 않는다고 생각 할 수있기 때문에, 차후 데이터 분석에서 제거하였다.

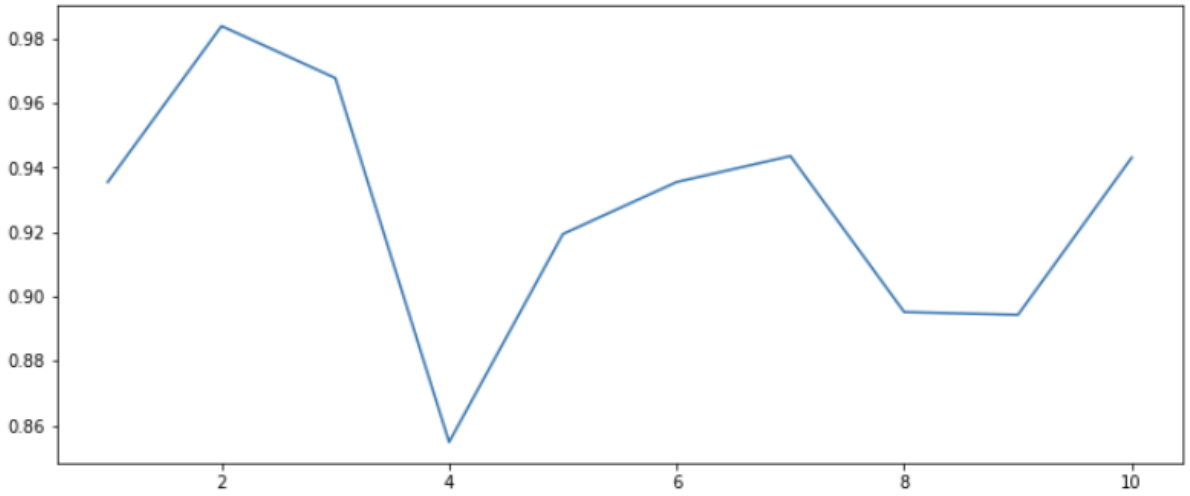


Fig 7. 10 - Cross Validation Random Forest

성별 변수가 제거된 후 같은 환경에서 Radnom Forest를 진행하였을시, Accuracy가 0.91에서 0.93으로 증가하였다. 10번의 Cross Validation을 통해 검증한 결과 (Fig 7.), 4번째 과정에서 Accuracy가 0.86으로 감소하였고, 2번째 과정에서는 0.98로 가장 높았다. 평균적으로 0.92를 유지하였으나, 그 차이가 0.1 이상으로 안정되었다고 말하기는 힘들다. Logistic Regression의 경우에도 같은 환경에서 Cross Validation을 진행한 결과 비슷한 양상을 보이며 전체적인 Accuracy는 RandomForest보다 낮았기 때문에 최종적으로 RandomForest를 선택하였다.

# 3. 머신러닝 방법 II

## 3.1 데이터 전처리

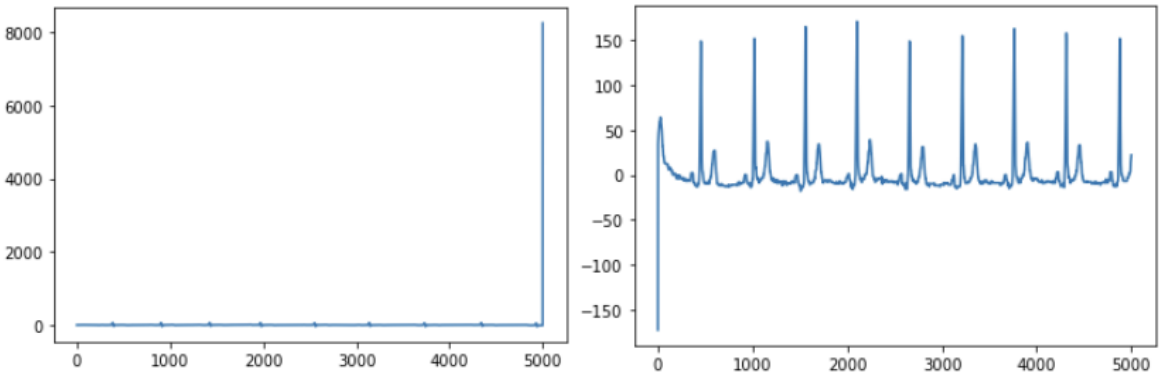


Fig 8. 이상치 관측 결과

12 ECG Lead Wave 데이터의 경우 몇몇 데이터의 첫번째, 마지막 값이 위의 그래프(Fig 8) 처럼 상대

적으로 높거나 낮아 전체적인 데이터의 분포에 큰 영향을 미치고 있다. 이에 따른 대처 방법으로는 첫번째, 마지막 값을 옆의 값으로 대체하는 방법과 제거하는 방법이 있을수 있다. 본 데이터의 경우, 5000개의 관측치가 있고, 첫번째와 마지막 2개의 데이터 제거가 크게 영향을 미치지 않을것이라 판단하여, 추후 분석에서는 제거되었다. 또한, 해당 데이터의 Class가 불균형하기 때문에, 불균형 문제를 해결하기 위해 SMOTE 기법을 사용하였다. 따라서, 최종 훈련에 사용될 데이터는 각각의 Class를 619개로 똑같이 맞춰준 데이터이다.

### 3.2 모델링: Neural Network

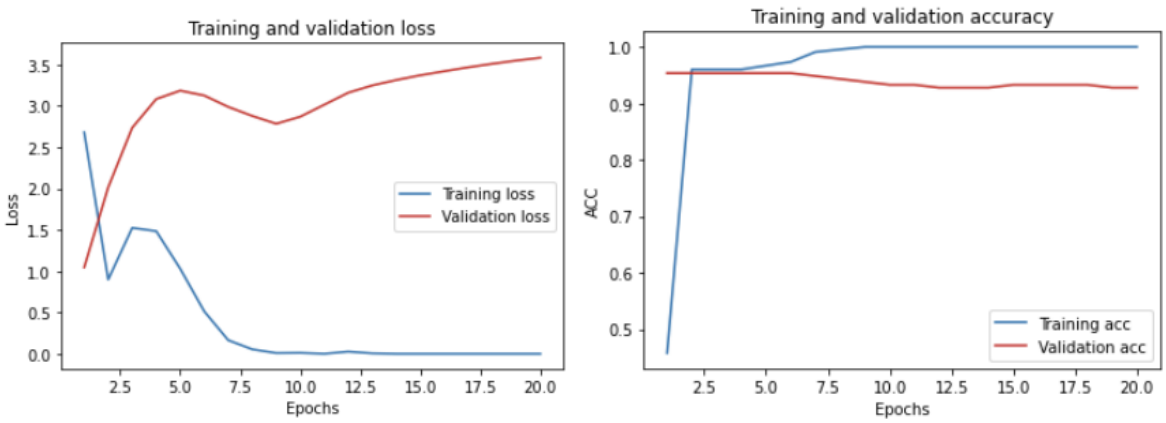


Fig 9. Training and Valdiation Accuracy and Loss

Neural Network 모델에서는 Loss function과 Metric은 각각 Binary Crossentropy와 Accuracy로 고정시킨 후, Layer의 개수, 각 Layer의 Activation Function 종류, Neural의 개수, Optimizer 를 변경하며 최적의 모델을 선정하였다. 전체 데이터는 0.3 비율로 Training과 Validation 데이터로 나눈 후, 몇 몇 Parameter를 변경하며 Validation의 Accuracy를 가장 높이면서 Validation Loss를 가장 낮추는 최적의 모델을 선정하였다. 최종 선정된 모델은 Layer가 6개이며, 각각의 layer는 128개의 Neuron을 가진다. 마지막 layer를 제외하고는, relu activation function을 이용하며, 마지막에는 두 가지 클래스를 분류하기 위해 sigmoid를 사용하였고 adam optimizer를 통해 최적화를 진행하였다. 결과적으로 20 epoch가 진행되면서 가장 최적의 상태일 때의 모델의 가중치를 저장하였으며, 이는 두번째 Epoch일 때, validation loss와 accuracy는 각각 4.3과 0.93이었다.

## 4. 최종 결과

### 4.1 최종 모델 선정

본 프로젝트에서는 총 세가지 모델이 학습되었다. Logistic Regression, Random Forest Classifier 그리고 Neural Network이다. Logistic Regression의 경우, 다른 두 모델에 비해 Accuracy가 현저히 낮았

다. Random Forest의 경우, 아직 Fine Tuning을 진행하지 않았기 때문에, 이를 통해서 parameter을 조정하여 좀 더 좋을 결과를 얻을 수 있을것으로 기대된다. 하지만, 전처리 과정 간 결측치를 채우는 과정에서 Raw Data만을 사용한 것이 아니기 때문에, 실제 테스트 데이터가 비슷한 성능을 보인다고 기대하기는 힘들다. 만약 테스트 데이터에도 비슷하게 결측치가 많은 상황이라면, 마지막 Neural Network 모델을 이용하는 것이 전체적으로 가장 좋은 성능을 낼 수 있을것으로 기대된다. 혹은, 두 가지 모델을 모두 사용하여 각 Class에 대한 확률을 계산하여, 더 높은 확률을 가지는 Class를 선택하는 Voting 작업을 진행하면 더욱 테스트 성능을 늘릴 수 있을것으로 기대한다.

### 4.2 테스트 결과

테스트 데이터는 총 285개가 존재하며, 테스트 데이터 내의 Xml 파일의 태그 저장 방식이 달라 테스트를 테스트1,테스트2,테스트3로 나누어 진행하였다. 테스트1에는 총 195개의 데이터가 저장되었다. Neural Network를 통한 테스트1 결과, ECG Lead 12개를 모두 사용한 Neural Network의 경우, Accuracy가 0.06이고 Loss가 81으로써, 훈련과정에서 과적합이 일어난것으로 판단하여, 총 3가지 가설로 새로운 모델을 시도하였다. 첫째로, ECG Lead가 너무 많아 데이터양을 줄이기 위해, 첫번째 ECG Lead 데이터만 사용하였다. 이후, 정규화를 시도하여 Lead의 수치를 전체적으로 동일하게 맞춰주었다. 마지막으로는, ECG Lead 데이터는 0~5000까지의 데이터지만 500씩 10번이 반복되어 측정되기 때문에, 0~500으로 데이터를 줄여주었다. 그러나, 모든 과정에서 Loss는 조금씩 줄었지만, Accuracy 역시 상승하지는 못하였다. RandomForest의 경우, 테스트1에 대해 Accuracy를 0.15를 가졌지만, 테스트 과정에서 결측치 데이터가 상당 부분 제거되었다. 또한, 테스트2 및 테스트3 데이터에는 해당 정보가 존재하지 않아 진행 테스트가 불가능하였다.

### 4.3 최종 결론

본 프로젝트에서 주요하게 사용된 기법은 SMOTE이다. SMOTE를 통해 훈련 데이터의 Class 불균형함을 해결함으로써, 각각 모델의 성능을 높일 수 있었다. 사용 된 모든 모델에 대해서 SMOTE 기법을 적용하지 않은 결과와 비교함으로써 SMOTE를 통해서 향상된 모델의 성능을 비교해 볼 수 있을 것이다. 모델링 과정에서는 과적합을 해결하는 문제가 가장 중요한 요소였다. Random Forest의 경우, 단순히 Tree, Node 등의 개수를 높임으로써, 훈련 데이터에 대해서는 정확도를 높일 수 있었으나, 이 결과가 테스트 성능에도 똑 같은 영향을 미치지는 못하였다. 또한, Neural Network도 layer의 개수, Neuron의 개수 등을 증가 시킴으로써 훈련 데이터의 정확도를 높였으나, 테스트에서도 같은 결과를 기대하기는 힘들었다. 따라서, 과적합을 최대한 해결하기 위해 Validation 데이터를 준비하여 각 훈련마다 Validation 데이터로 한번 더 테스트를 진행함으로써 해당 모델의 가중치의 유효성을 입증하여 향후 모델의 성능을 발전시킬수 있었다. 하지만, 최종 테스트 결과의 과적합을 해결하지는 못한 모습이 보였다.