1. WESAD 데이터셋 설명
2. 전처리 과정 설명
    -> Mean Heart Rate
    -> STD Heart Rate
    -> TINN
    -> HRVindex
    -> #NN50
    -> pNN50
    -> Mean HRV
    -> STD HRV
    -> RMS HRV
    -> Mean Fourier Frequencies
    -> STD Fourier Frequencies
    -> Sum PSD components

3. 데이터셋 생성
    - k-fold dataset
    - ECG 데이터의 person dependency 해결 방법
    - balancing dataset

4. 모델 생성
    - 모델 레이어 설명
5. 모델 평가
    - confusion matrix
    - accuracy, f1 score, …..
6. API 만들기
    - API 만드는 과정(pickle, json 통신, ,,,,)

1. WESAD 데이터셋 설명

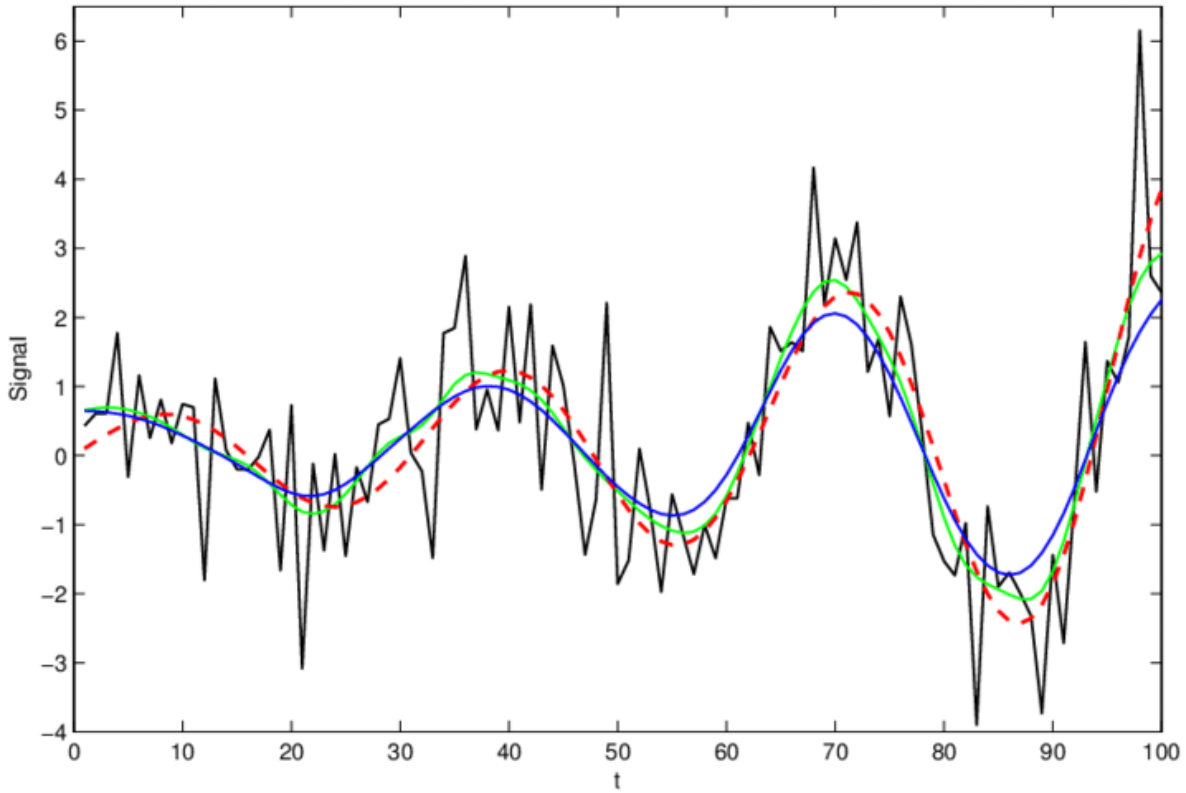
- WESAD 데이터셋은 총 1번부터 17번까지의 사람들의 두시간짜리 ECG 데이터가 pkl 파일로 저장되어 있다.
- 이 중에서 1번, 12번 피험자의 센서에 문제가 있어 데이터셋에서 제외되었다 .

2. 전처리 과정 설명

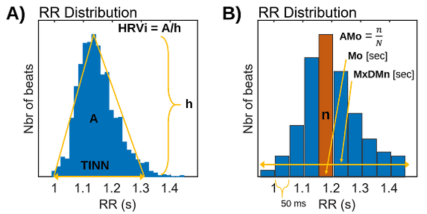
- ECG 데이터는 time domain, frequency domain 두 가지로 분석할 수 있다.

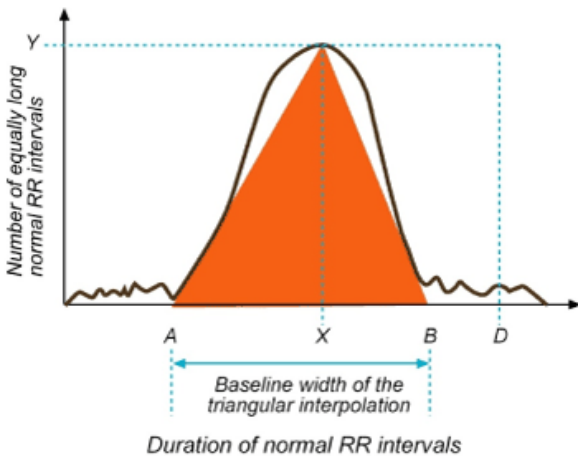
1. Time-domain
    1. Statistical – 통계적 방법.
        1. NN50: 연속적인 NN 간격의 차이가 50ms를 초과하는 NN간격의 개수
        2. pNN50: NN50 개수를 총 NN 수로 나누어 구함. 주로 부교감 신경의 활동을 반영.
    2. Geometric – 기하학적 방법. NN interval의 시퀀스는 기하학적 패턴으로 표현될 수 있다.
        1. TINN: NN간격 히스토그램의 밑변 너비
2. Frequency domain

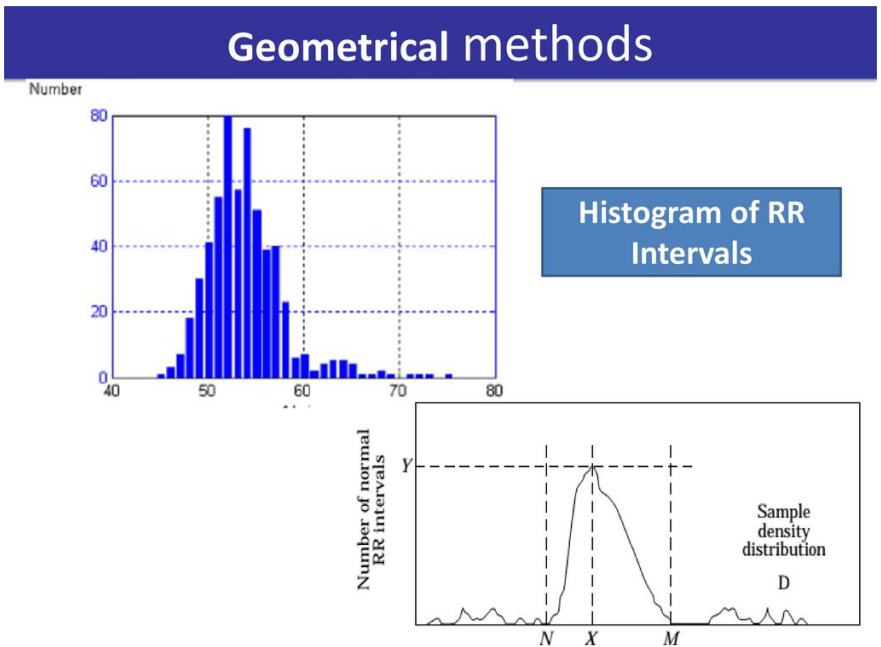
- NN interval 측정을 위한 peak detection 을 위한 signal smoothing시발이게맞누
직접 측정하는 ECG데이터에서는 일차적으로 피크 감지 기능이 있지만 ,
WESAD 데이터셋은 그저 시계열 데이터로 존재하기 때문에 직접 peak를 감지해야 했다.
따라서 peak 감지 이전에 정확도 향상을 위해 먼저 convolution을 이용하여 signal을 smoothen 한 후에 peak를 감지하도록 하였다.
이때 Window size는 5로, 각 window마다 convolution을 이용하여 평균 값을 구한다.
이후에 특정 값(baseline) 이상인 index를 피크 인덱스로 반환하는 방식으로 peak 감지 기능을 구현하였다.



분포 히스토그램을 삼각형으로 근사화: TINN[RR(또는 NN 간격) 히스토그램의 삼각 보간] 측정에 대해 RR 간격(히스토그램 카운트)의 이산 분포의 삼각형 보간법 사용 : TINN = B– A [12] (그림 1).
가우스 분포의 최대값인 추축의 오른쪽과 왼쪽에 각각 2개의 부분으로 나뉜다.
TINN 점수 계산은 WESAD Dataset 논문에 정의되어 있다.
그에 따르면, NN interval의 분포를 가우시안 분포에 근사화하여 구한 확률 밀도 함수로부터 triangular interpolation을 구하는 것이 필요하다.
triangular interpolation은 가우시안 분포의 최대값에서 만나고 x축의 전반부의 N과 x축 후반부의 M에서 x축을 교차하는 2개의 선으로 정의된다.





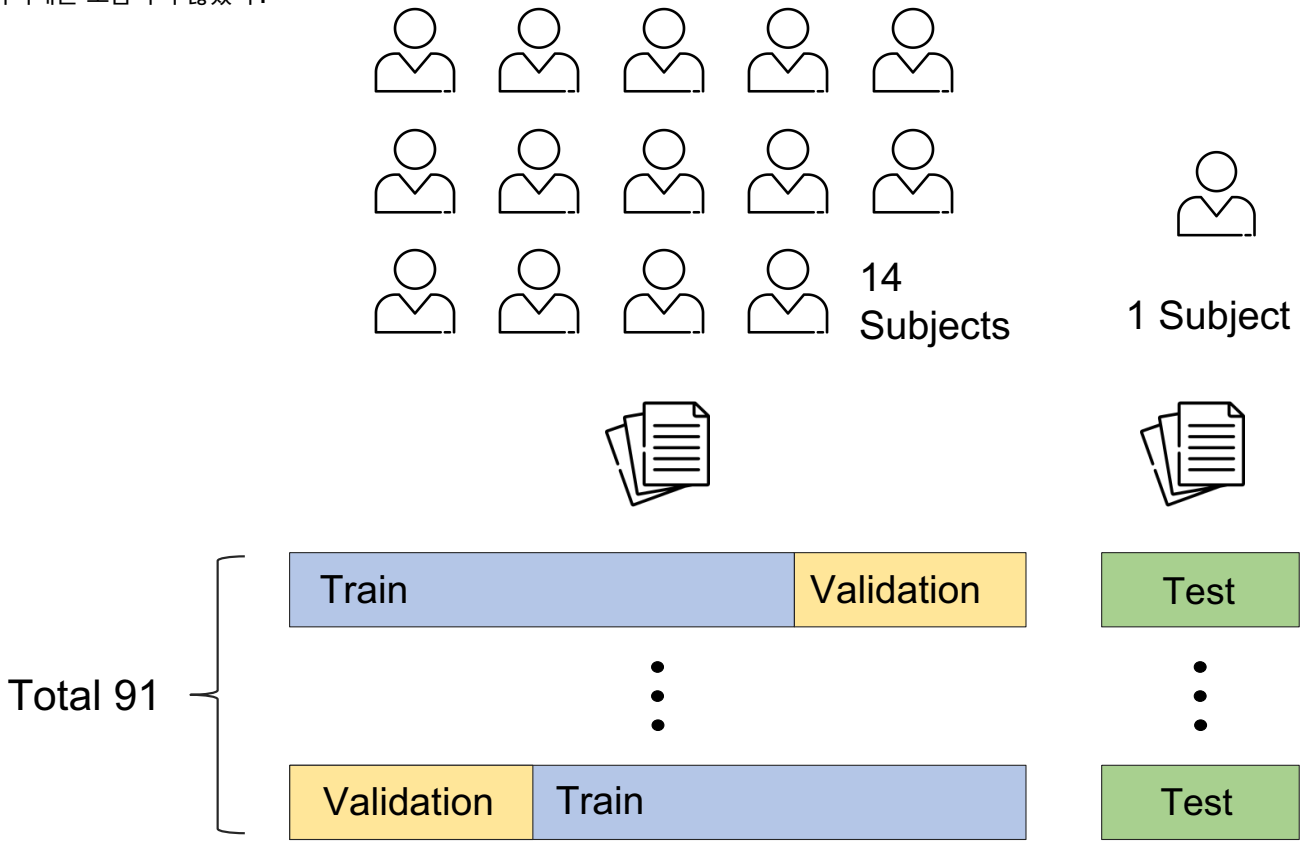


3. 데이터셋 생성

- K-fold cross validation
- ECG 데이터의 person dependency 해결 방법
- Balancing dataset

- K-fold cross validation

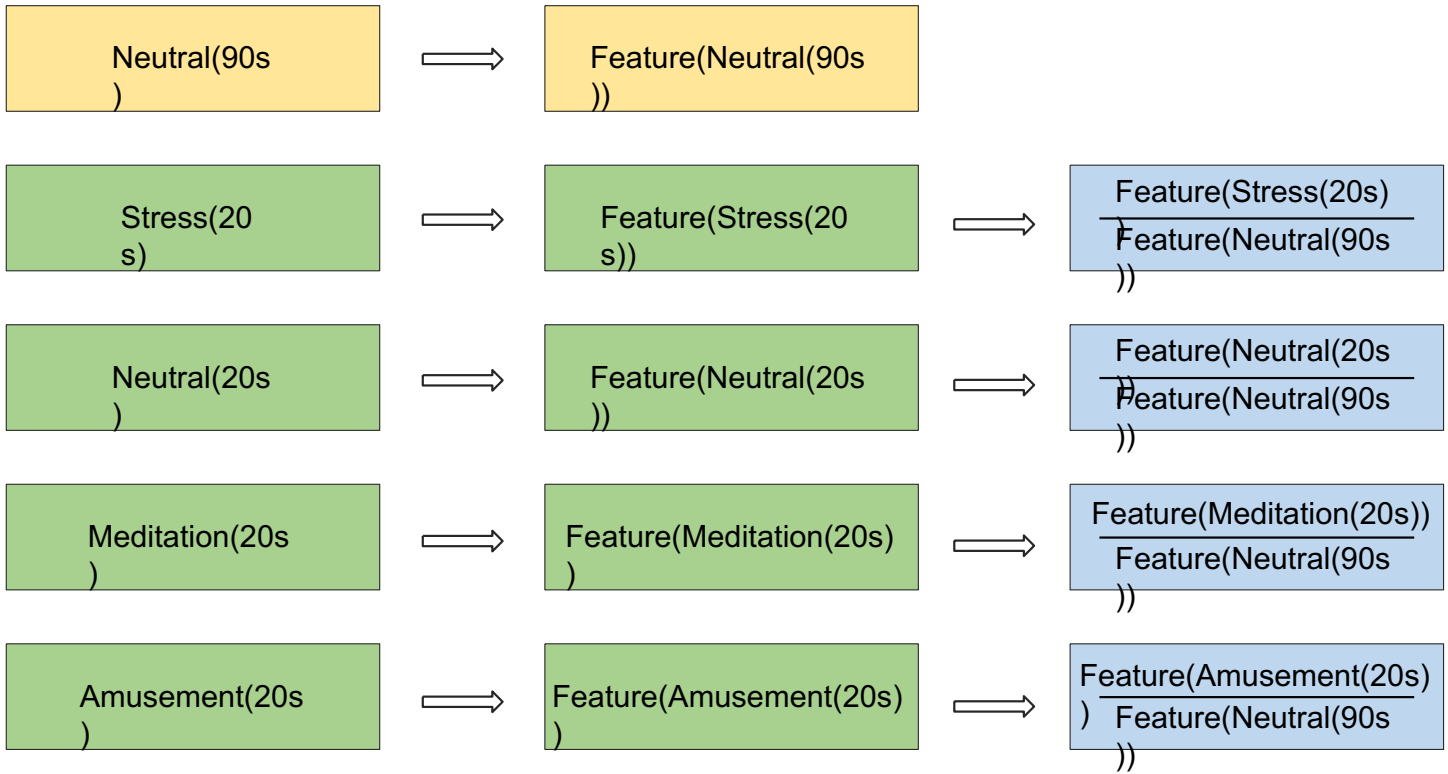
: Overfitting을 방지하기 위해 15개의 데이터셋을 Training set, validation set, testing set으로 나누었다.
k-fold cross validation(training:12, validation:2) 을 사용하였고 테스트 데이터로는 17번 사용자의 데이터를 선택했으며 학습 데이터에는 포함하지 않았다.



- ECG 데이터의 person dependency 해결 방법

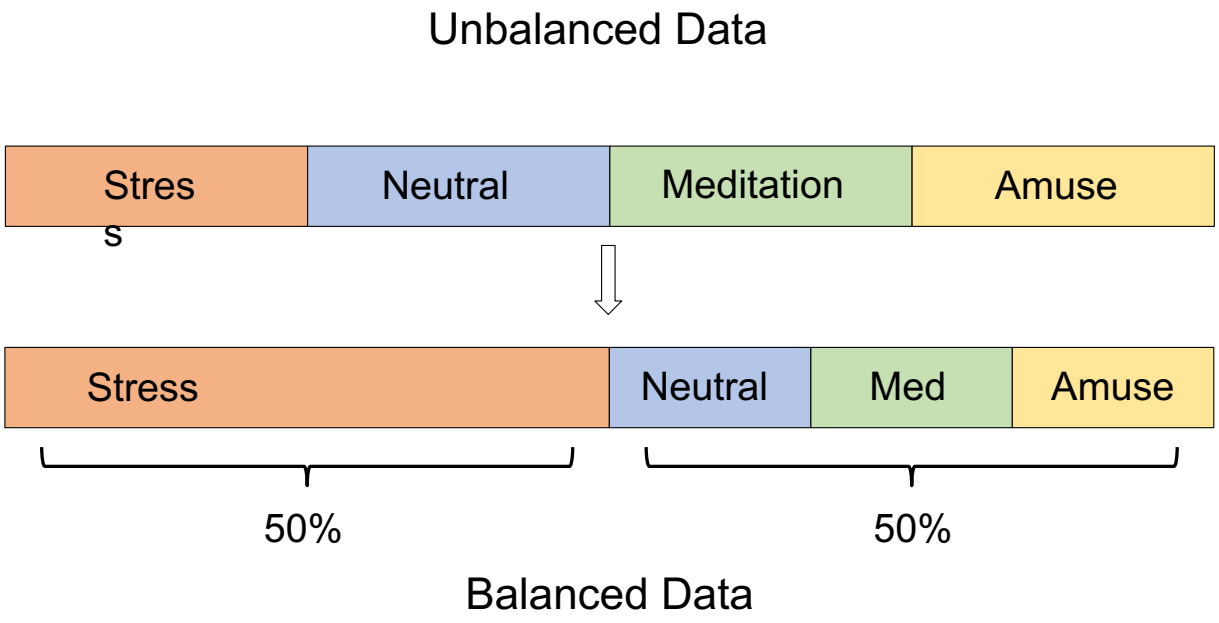
ECG 데이터는 개인차가 매우 크다는 문제가 있다 .

→ neutral 90초 데이터로 추출한 feature로 모든 20초 샘플의 feature를 나누어서 이 문제를 해결하였다 .



- Balancing dataset

데이터셋에서 이상치를 처리한 이후,
Stress/non-stress 이진 분류 성능을 향상시키기 위해 Stress와 non-stress 비율을 50:50으로 맞추었다.



4. 모델 설명

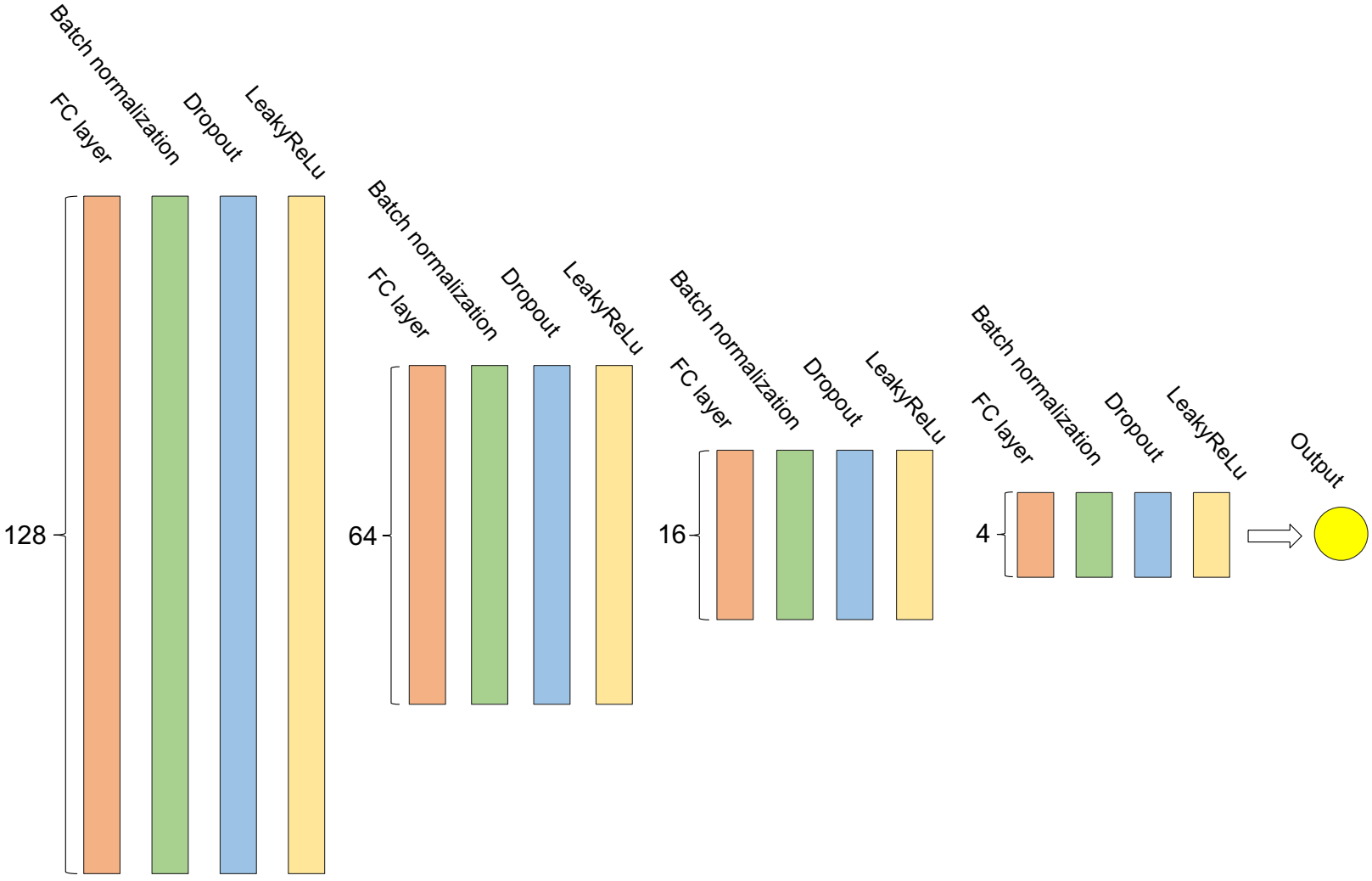
- 레이어 설명

레이어는
저장된 가중치(weight)와 편향(bias)을 사용하여 입력에 선형 변환 (linear transformation)을 적용하는 nn.Linear 레이어,
배치 정규화,
과적합을 방지하기 위해 랜덤 확률로 뉴런을 제거하는 dropout,
활성화 함수 순이다.
마지막에는 이진 분류를 위해 sigmoid 함수를 사용하였다.

4. 모델 설명

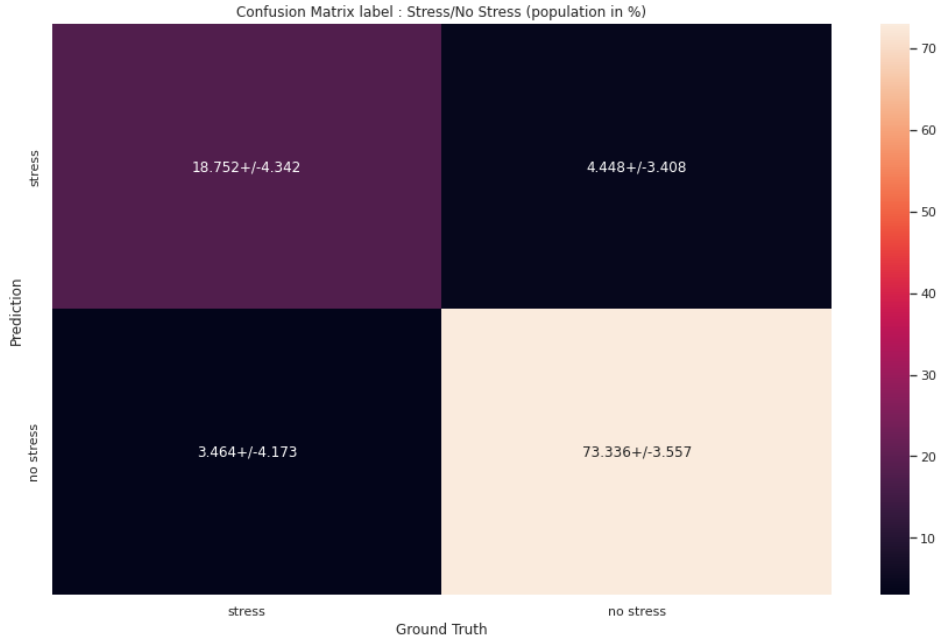
```
class ClassifierECG(nn.Module):
    def __init__(self, ngpu):
        super(ClassifierECG, self).__init__()
        self.ngpu = ngpu
        self.nnECG = nn.Sequential(
            nn.Linear(12,128,bias=True),
            nn.BatchNorm1d(128),
            nn.Dropout(0.5),
            nn.LeakyReLU(0.2),
            nn.Linear(128,64,bias=True),
            nn.BatchNorm1d(64),
            nn.Dropout(0.5),
            nn.LeakyReLU(0.2),
            nn.Linear(64,16,bias=True),
            nn.BatchNorm1d(16),
            nn.Dropout(0.5),
            nn.LeakyReLU(0.2),
            nn.Linear(16,4,bias=True),
            nn.BatchNorm1d(4),
            nn.Dropout(0.5),
            nn.LeakyReLU(0.2),
            nn.Linear(4,1,bias=True),
            nn.Sigmoid()
        )
        self.nnECG.apply(init_weight)

    def forward(self, input):
        return self.nnECG(input)
```

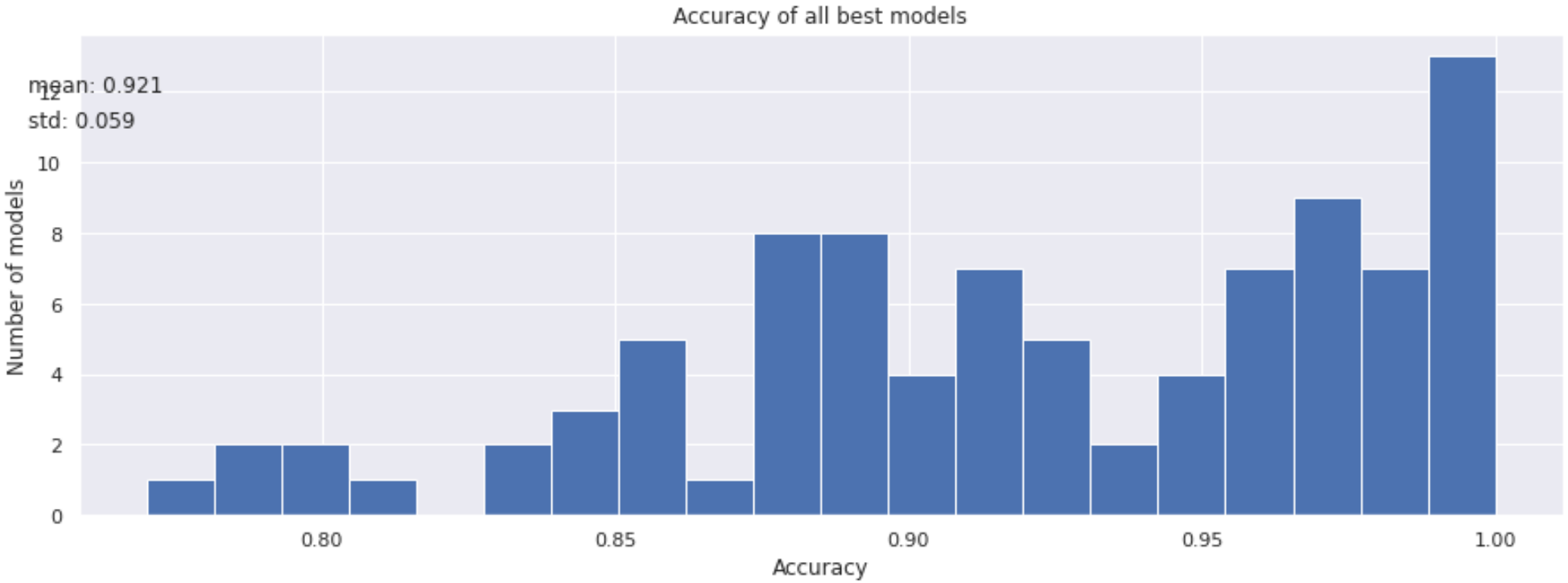

FC layer
Batch normalization
Dropout
LeakyReLu
128
FC layer
Batch normalization
Dropout
LeakyReLu
64
FC layer
Batch normalization
Dropout
LeakyReLu
16
FC layer
Batch normalization
Dropout
LeakyReLu
4
Output

5. 모델평가

- Confusion Matrix

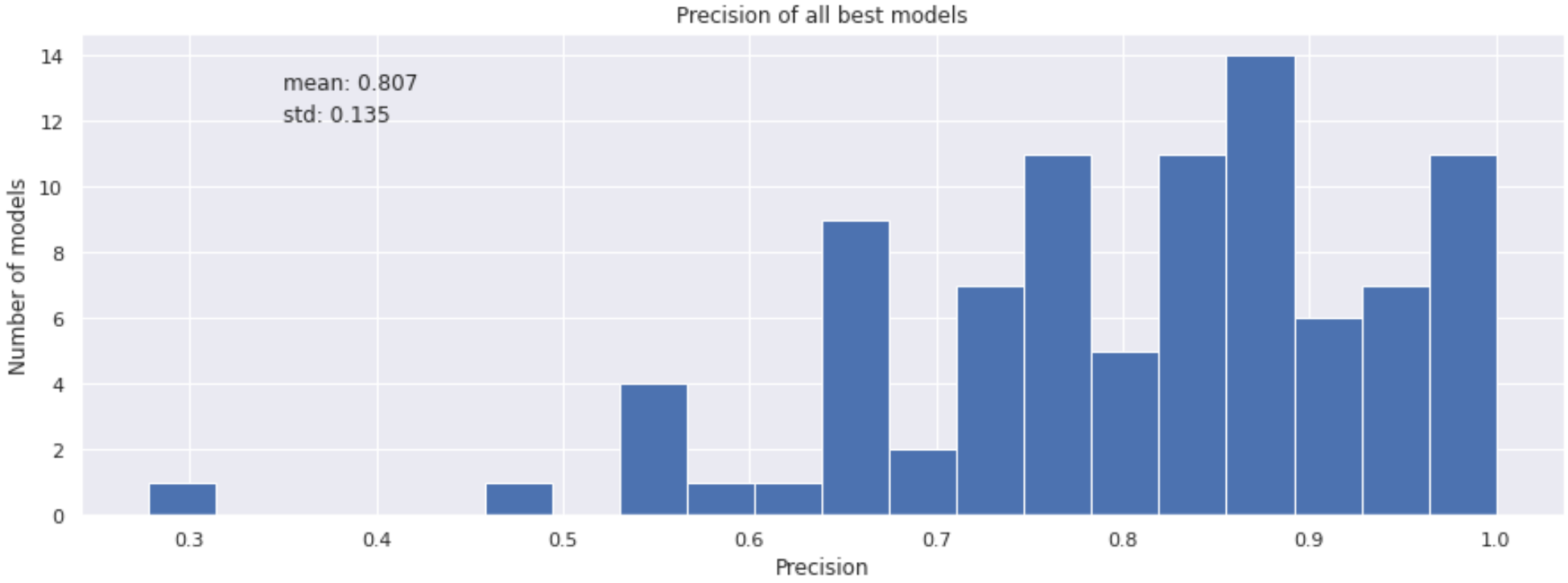


5. 모델평가

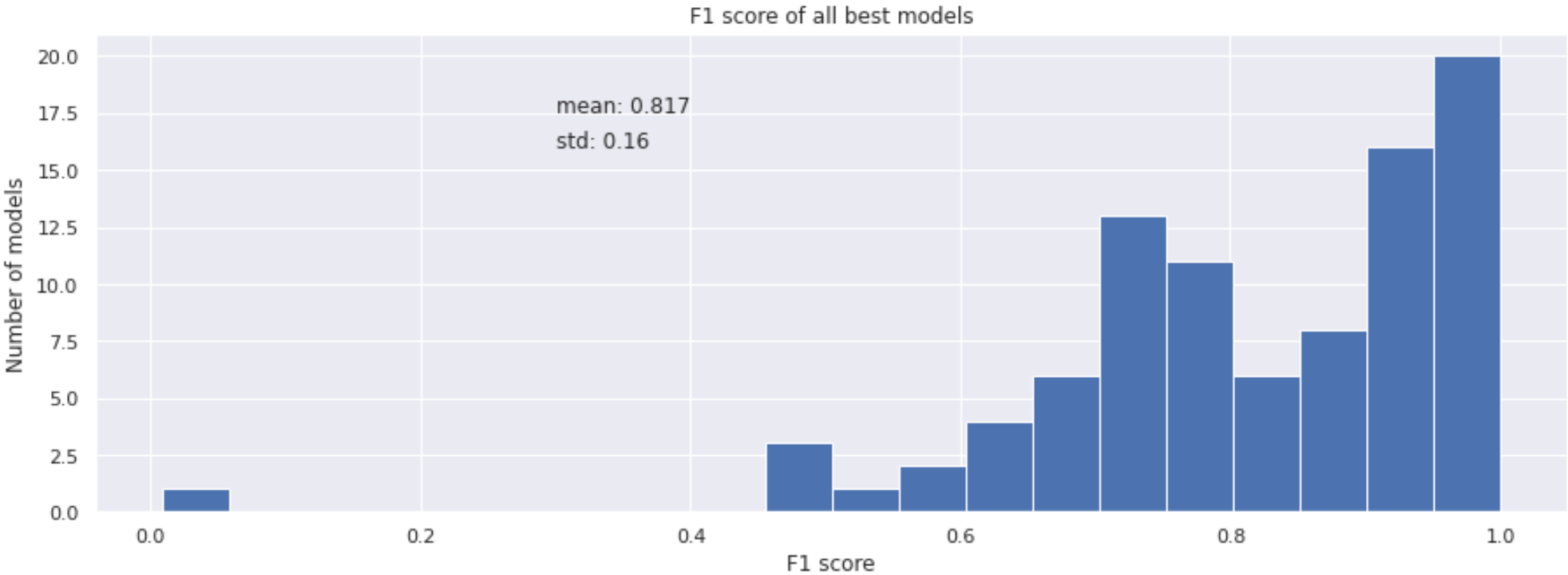

Accuracy of all best models
mean: 0.921
std: 0.059
Number of models
0
2
4
6
8
10
12
0.80
0.85
0.90
0.95
1.00
Accuracy

5. 모델평가

정밀도

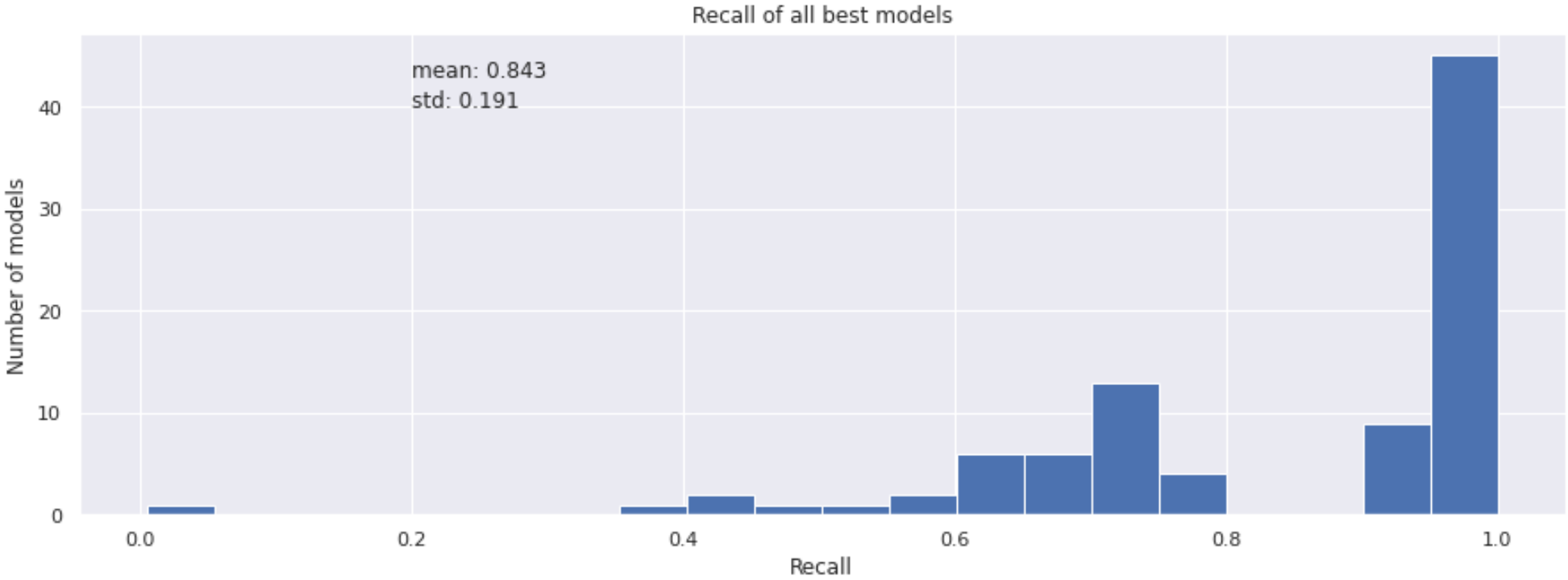

Precision of all best models
mean: 0.807
std: 0.135
Number of models
Precision
0
2
4
6
8
10
12
14
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

5. 모델평가


F1 score of all best models
mean: 0.817
std: 0.16
Number of models
F1 score
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0.0
0.2
0.4
0.6
0.8
1.0

5. 모델평가

재현율


Recall of all best models
mean: 0.843
std: 0.191
Number of models
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0
Recall