다변량분석 4번째 과제

Assignment 4

Decision tree for classification

학과 산업경영공학부

학번 2016170863

이름 추창욱

<voice.csv 목소리 성별 맞추기>

지난 Assignment 2에서 제가 사용한 데이터는 목소리로 성별을 맞추기 위한 데이터였습니다. 이 데이터를 사용하기 위해 Preprocessing 과정을 거쳤는데, 총 20개의 설명변수와 한 개의 타겟 변수로 이루어졌다는 것을 확인 할 수 있었습니다. 설명 변수들 중, 사용하지 않는 설명 변수는 없었으며, input 변수들을 scaling 과정을 한번 거쳐 준 후, target 변수는 factor화 시켜 각각 지정해주었습니다. 이후 이 둘을 하나의 데이터 프레임으로 만든 후 train 과 test 셋으로 나누어 주었습니다.

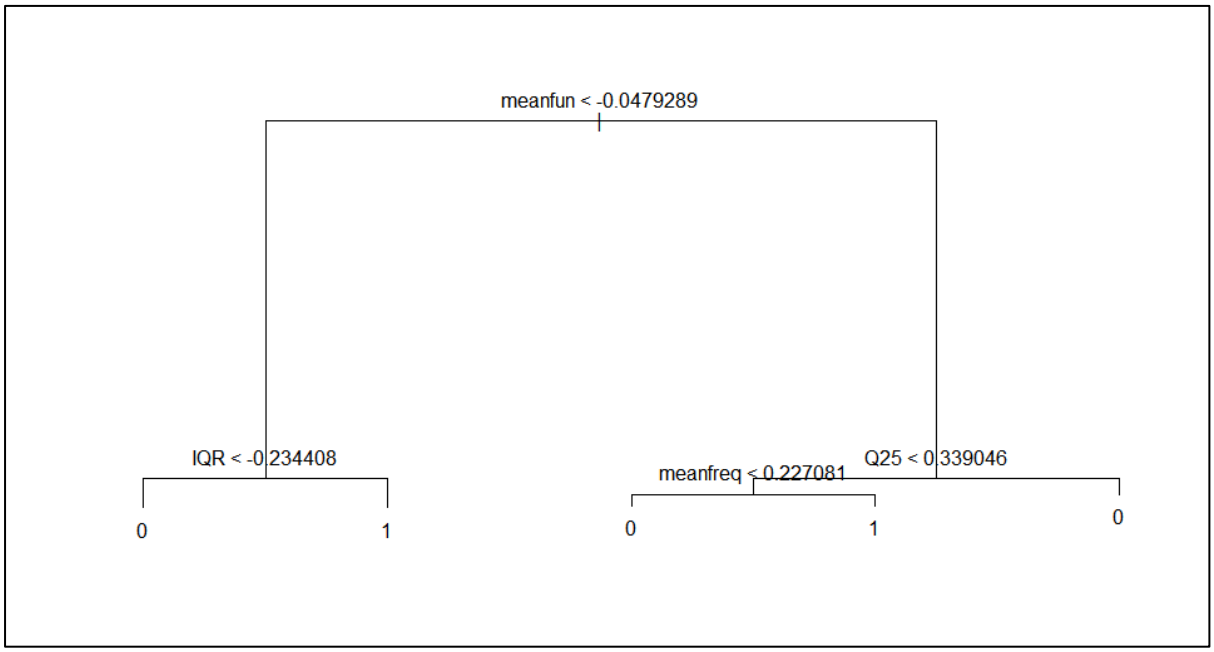
[Q2] 실습 시간에 사용한 "tree" package 를 사용하여 Classification Tree 를 학습한 결과물을 Plotting 하고 이에 대한 해석을 수행하시오. 또한 해당 Tree 를 pruning 을 수행하지 않은 상태에서 Test dataset 에 대한 분류 성능을 평가하시오.

우선 저는 train : test를 60 : 40의 비율로 나누어 준 후, 2번 문제를 진행하였습니다. 그 이유는 데이터를 나누는 데 있어 training 데이터가 너무 많으면 tree는 과적합이 쉽게 일어날 수도 있을 것 같다는 생각에 최대한 비슷한 비율로 하기 위해 적절한 값이 60:40이라고 생각하였습니다.

이에 대한 결과로 얻을 수 있었던 summary는 다음과 같습니다.

```
Classification tree:
tree(formula = voice_target ~ ., data = voice_trn)
Variables actually used in tree construction:
[1] "meanfun"  "IQR"      "Q25"      "meanfreq"
Number of terminal nodes:  5
Residual mean deviance:  0.2064 = 391.3 / 1896
Misclassification error rate: 0.03156 = 60 / 1901
```

세번째 줄에서 알 수 있듯이 실제로 이 데이터에서 트리를 만드는데 이용된 변수는 총 4개의 변수가 있다는 것을 확인 할 수 있었습니다. Number of terminal nodes가 5라는 것을 통해 leaf node가 5개라는 것을 알 수 있었습니다. 이를 plotting 해보면 다음과 같은 결과를 얻을 수 있었습니다.



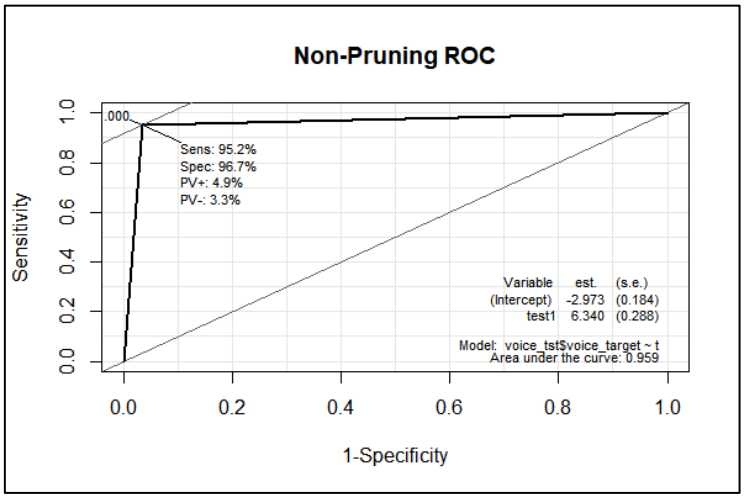
가장 먼저 사용된 변수는 meanfun으로 meanfun이 -0.0479289보다 작으면 왼쪽으로 가게 되고, 그 중에서 IQR이 -0.234408보다 작으면 여성, 높으면 남성과 같은 규칙으로 트리가 구성되어 있다는 것을 확인할 수 있었습니다.

성능을 평가하기 위해 이전에 작성한 Performance mat을 돌려본 결과는 다음과 같았습니다.

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Non-Pruning	0.9515625	0.9666667	0.9665072	0.9589582	0.9590057	0.9590551

TPR, 실제로 남성의 목소리(1)를 남성(1)이라고 예측한 비율인 TPR이 95%로 꽤나 높게 나왔고, Precision 같은 경우는 모델이 남성이라고 예측한 비율 중에서 실제로 맞춘 비율로서 96.6% 꽤나 높은 정확도를 나타낸 걸 확인 하였습니다. TNR은 여성을 여성이라고 맞게 예측한 비율로 96.6%이고, 단순 정확도는 95.9%, 균형 정확도는 95.9%, F1-measure은 95.9%라는 것을 확인하였습니다.

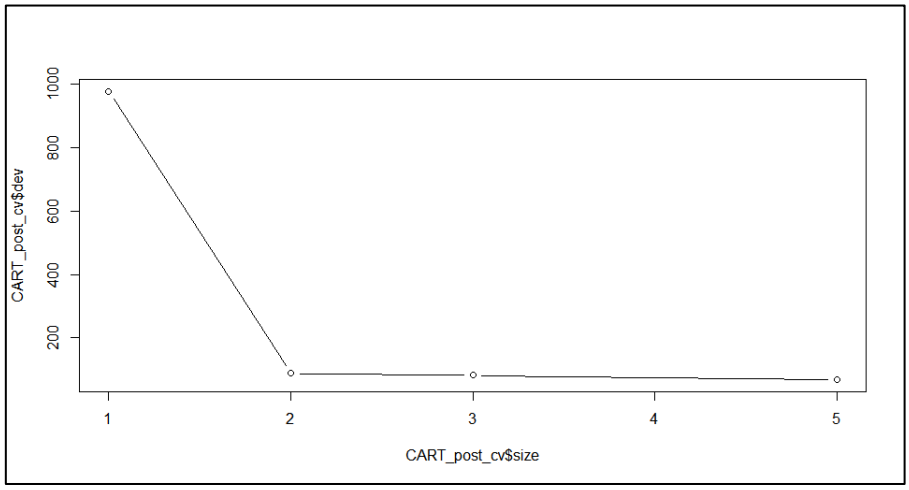
AUROC 그래프는 다음과 같이 나왔다는 것을 확인 할 수 있었습니다.



Area under the curve가 0.959가 나왔다는 점을 숙지하며 다음으로 넘어가도록 하겠습니다.

[Q3] 앞에서 생성한 Tree에 대해서 적절한 Post-Pruning을 수행한 뒤 결과물을 Plotting하고 이에 대한 해석을 수행하시오. Pruning 전과 후에 Split에 사용된 변수는 어떤 변화가 있는가? Test dataset에 대한 분류 성능을 평가하고 [Q2]의 결과와 비교해보시오.

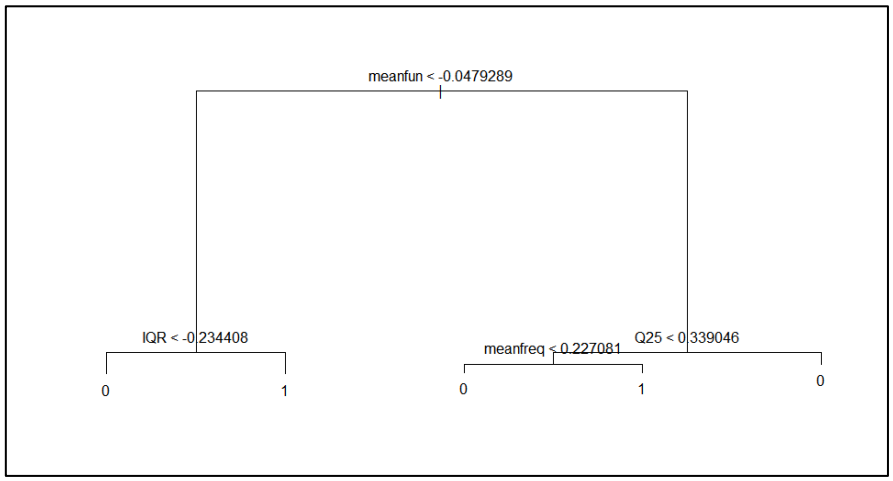
플롯팅한 결과물을 먼저 확인해보자면 다음과 같았습니다.



y축은 불순도이므로 처음 스플릿이 되자마자 급격하게 떨어지는 것을 확인할 수 있었고, 그후로 큰 변화를 보이지 않고 있다는 것을 확인하였습니다. 오른쪽 표를 통해 위 그래프에 대한 더 상세한 정보를 확인 할 수 있었습니다. 사이즈가 1일때는 979였으나 이후 사이즈가 2가 되었을 때 급격하게 87로 줄어든 것을 확인 하였고, 이후 미세하게 낮아지다가 사이즈가 5일 때 다시 한번 67로 낮아지는 것을 확인 할 수 있었습니다. 이를 통해 현재 5개의 리프노드를 가지고 있을 때 불순도는 67로 가장 낮으니까 최적의 모델이겠구나 라고 생각하였습니다.

```
$size
[1] 5 3 2 1

$dev
[1]  67  81  87 979

$k
[1]  -Inf   7.5  10.0 859.0

$method
[1] "misclass"

attr(,"class")
[1] "prune"
[2] "tree.sequence"
```

5로 설정을 하여 다시 plot을 그려본 결과 트리의 모습은 다음과 같았습니다.



이는 pruning을 하기 전과 동일한 결과값이 나왔다는 것을 확인 할 수 있었습니다. 사이즈의 변화가 생기지 않았기 때문입니다. 이에 따라 사용된 변수 또한 동일하다는 것을 확인 할 수 있었습니다.

Test 데이터 셋에 대한 분류 성능을 파악하기 위해서 confusion matrix와 이전에 짜놓은 performance mat에 넣어 돌려본 결과는 다음과 같이 나왔다는 것을 확인 할 수 있었습니다.

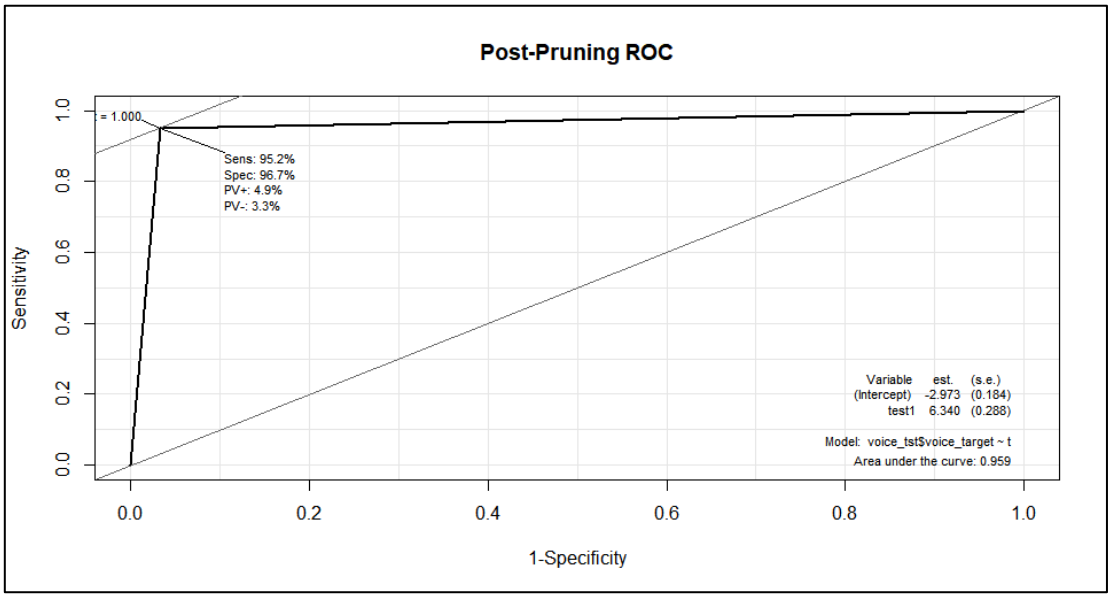
	CART_post_pray	
	0	1
0	606	21
1	31	609

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Non-Pruning	0.9515625	0.9666667	0.9665072	0.9589582	0.9590057	0.9590551
Post-Pruning	0.9515625	0.9666667	0.9665072	0.9589582	0.9590057	0.9590551

TPR 실제로 남성의 목소리(1)를 남성(1)이라고 예측한 비율인 TPR이 95%로 꽤나 높게 나왔고, Precision 같은 경우는 모델이 남성이라고 예측한 비율 중에서 실제로 맞춘 비율로서 96.6% 꽤나 높은 정확도를 나타낸 걸 확인 하였습니다. TNR은 여성을 여성이라고 맞게 예측한 비율로 96.6%이고, 단순 정확도는 95.9%, 균형 정확도는 95.9%, F1-measure은 95.9%라는 것을 확인하였습니다.

이 역시도 위에서 언급한 바와 같이 pruning을 하지 않았을 때와 같은 결과값이 나올 수 밖에 없다는 것을 알 수 있었습니다. 사이즈는 변하지 않았고, 이에 따라 split하는데 사용된 변수들 또한 변하지 않았기 때문입니다.

AUROC 관점에서 비교를 해본 결과는 다음과 같았습니다. 우선 ROC 커브를 그려본 결과는 다음과 같았습니다.



AUC의 값은 0.959라는 값이 나왔으며, 이 또한 non-pruning이랑 같은 결과가 나왔다는 것도 확인 할 수 있었습니다.

[Q4] "party" package를 사용하여 다음 조건에 맞는 Pre-pruning을 수행하여 가장 최적의 min_criterion, min_split, max_depth 값을 찾아보시오.

우선, pre-pruning 을 하기 전 이번에는 60%의 데이터를 이전 non과 post-pruning 과정을 거쳤을 때와 동일한 비율로 training을 시키기 위해 먼저 trainset은 전체 데이터의 50%로 생성하였고, 나머지 데이터의 20%는 검증용 데이터, 나머지는 test용 데이터로 하였습니다. 결론적으로는 공평한 비교를 위해 이전 모델에서 분리한 train을 위한 데이터와 test를 위한 데이터의 비율이 같아지게 하였습니다. 단순히 60%를 training 으로 이용하고자 한 이유는 너무 많은 데이터의 수를 training으로 하면 과적합이 발생할 것 같다는 생각이 들어서 였으며, 이전 모델과 새로운 모델을 비교할 때, 동일한 비율로 train과 test를 나누는 것이 모델의 성능을 파악하기 더 용이 할 거라고 생각하여 위와 같이 데이터를 나누게 되었습니다.

```
# tree parameter settings
min_criterion = c(0.5, 0.6, 0.7, 0.8, 0.9, 0.99)
min_split = c(10,20,30,40,50,100)
max_depth = c(0, 5, 10, 15, 20)
```

저는 다음과 같은 후보 값을 주어 최적의 하이퍼 파라미터를 찾고자 하였습니다. 우선 하이퍼 파라미터를 다음과 같이 설정하기 위해서는 각각에 대한 이해가 필요하다고 생각하기에 아주 간략하게 설명을 하고 넘어가도록 하겠습니다.

Min_criterion: Party package는 deviance, gini index와 같은 것을 사용하지 않고 split한 이후에 그 둘 사이의 불순도의 감소가 얼마나 통계적으로 유의미하냐 를 test를 수행하게 되는데 이때, 각각의 값들은 예를 들어, 0.8이라면 80% 이상이면 허용을 해주자라는 조건을 의미합니다.

Min_split: 이 기준은 지금 우리가 대상으로 하는 영역에 최소한 관측치가 몇 개 있어야 하는가를 의미하는 하이퍼 파라미터로서 최소한 이정도의 관측치가 있어야 split의 후보군으로 사용하겠다는 조건을 보여줍니다..

Max_depth: split을 최대 몇 번까지 할 것 인가를 의미하는 조건이라고 볼 수 있을 것입니다.

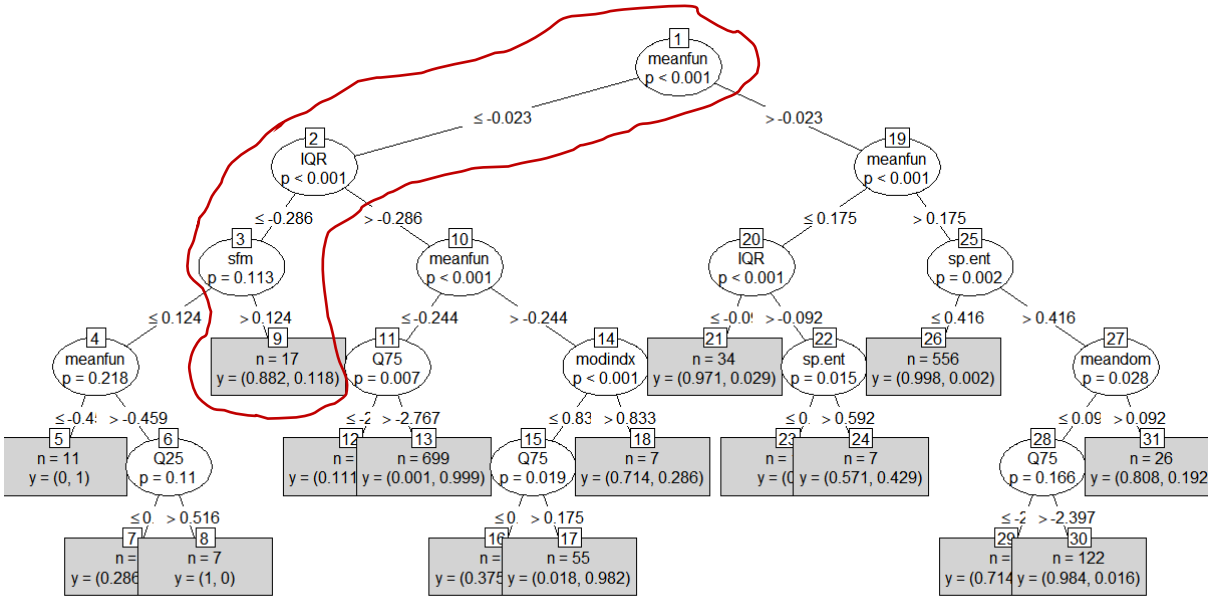
Iteration을 돌려서 최적의 하이퍼 파라미터를 찾기 위한 코드를 돌려보았습니다. 이때 최적의 하이퍼 파라미터라는 것을 정하는 평가지표는 AUROC였으며 결과는 다음과 같이 나오는 것을 확인 할 수 있었습니다.

	min_criterion	min_split	max_depth	TPR	Precision	TNR	ACC	BCR	F1	AUROC	N_leaves
[1,]	0.50	10	5	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9852122	16
[2,]	0.60	10	5	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9852122	16
[3,]	0.70	10	0	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9852122	16
[4,]	0.70	10	5	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9852122	16
[5,]	0.70	10	10	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9852122	16
[6,]	0.70	10	15	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9852122	16
[7,]	0.70	10	20	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9852122	16
[8,]	0.50	10	0	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9851326	17
[9,]	0.50	10	10	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9851326	17
[10,]	0.50	10	15	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9851326	17
[11,]	0.50	10	20	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9851326	17
[12,]	0.60	10	0	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9851326	17
[13,]	0.60	10	10	0.9622642	0.9745223	0.9746835	0.9684543	0.9684539	0.9683544	0.9851326	17

현재 제가 가지고 있는 데이터가 그리 복잡하지 않은 데이터 셋이라는 점에서 가장 높은 AUROC가 나오는 값이 여러 개 있다는 것을 확인 할 수 있었습니다. 가장 높은 AUROC값이 나왔을 때의 값은 0.9852122였으며 이때 가장 위에 있는 하이퍼 파라미터는 위에 언급한 순서대로 0.5, 10, 5일 때라는 것을 확인 할 수 있었습니다.

[Q5] 최적의 결정나무의 Plot을 그리고, 대표적인 세 가지 규칙에 대해서 설명해보시오.

위에서 선정된 최적의 결정나무를 가지고 의사결정나무 규칙을 plotting해본 결과는 다음과 같이 나오는 것을 확인하였습니다.



하나의 과정만을 살펴 보도록 하겠습니다. 가장 처음으로 split을 하는데 사용된 분할 변수는 meanfun이라는 것을 볼 수 있습니다. 만약 meanfun의 밸류가 -0.023보다 낮다면 다음으로 보게 되는 변수는 IQR이라는 것을 알 수 있습니다. IQR의 값이 -0.286보다 작은 데이터들은 stm을 보게 되는데 여기서 0.124보다 높은 값을 지닌 데이터를 살펴 보았더니 0 즉, 여성일 확률이 0.882였고, 남성일 확률이 0.118이였으며 이에 해당하는 데이터들은 총 17개가 있었다 라는 정보를 화인 할 수 있었습니다. 이런 식으로 여러 규칙들이 존재하고 이를 통해서 여성의 목소리 인지 남성의 목소리인지를 어떻게 구별하는지에 대한 규칙을 확인 해볼 수 있었습니다.

[Q6] [Q4]에서 선택한 하이퍼라라미터 조합을 이용하여 Training Dataset과 Validation Dataset을 결합한 데이터셋을 학습한 뒤, Test Dataset에 적용해보고 분류 성능을 평가하시오.

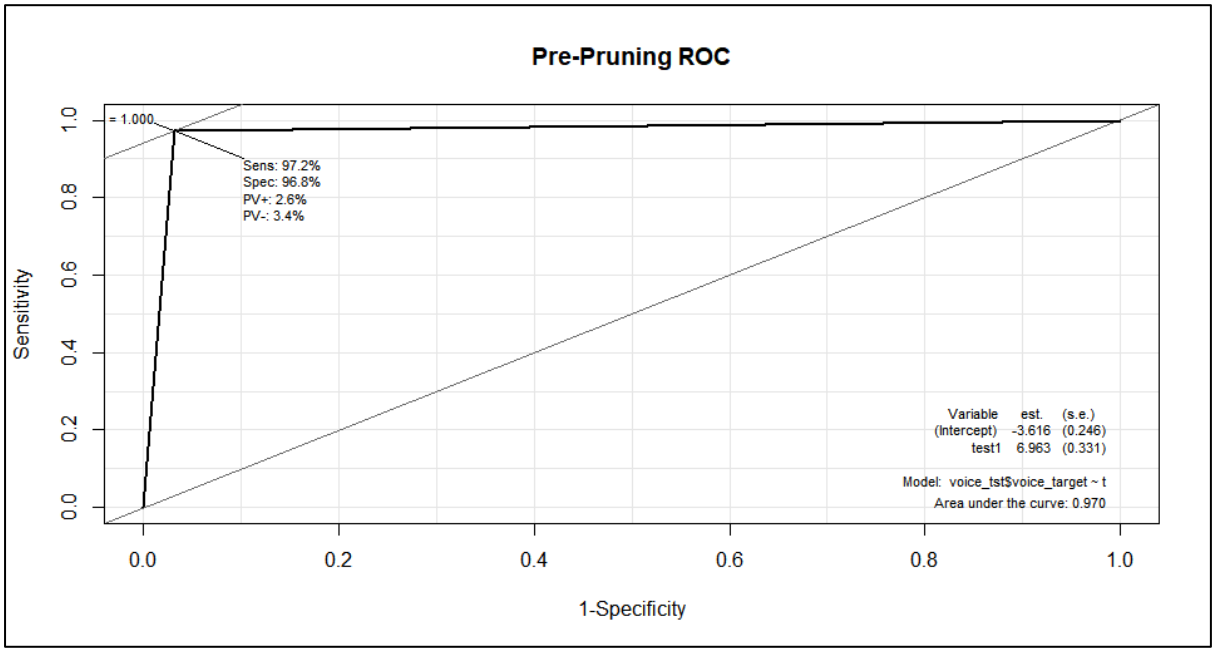
제가 선택한 하이퍼파라미터 조합을 이용하여 모델을 돌려본 결과 다음과 같은 confusion matrix와 성능 평가 지표들이 나왔다는 것을 확인 할 수 있었습니다.

CART_pre_prediction		
	0	1
0	632	21
1	17	597

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Non-Pruning	0.9515625	0.9666667	0.9665072	0.9589582	0.9590057	0.9590551
Post-Pruning	0.9515625	0.9666667	0.9665072	0.9589582	0.9590057	0.9590551
Pre-Pruning	0.9723127	0.9660194	0.9678407	0.9700079	0.9700741	0.9691558

이전의 모델(non-pruning과 post-pruning은 같은 결과가 나왔기에 하나로 묶겠습니다)과 비교해 보았을 때, confusion matrix를 확인해보면, 남자인데 여자의 목소리라고 오 분류 한 개수는 줄었지만, 여성인데 남성이라고 오분류한 개수는 조금 늘었다는 것을 확인 해볼 수 있었습니다. 이에 따라, TPR 값은 증가 하였으나 Precision은 감소한 모습을 보였고, 그럼에도 TNR, Accuracy, BCR, F1-measure에서 더 높은 수치를 가지므로 성능이 개선되었다는 결론을 내릴 수 있었습니다.

Voice test에 대해 모델을 돌려 AUROC관점에서 비교를 해보자면, 우선 AUROC의 모양은 다음과 같이 나왔습니다.



높은 결과가 나왔다는 것을 확인 할 수 있었으며 이전에 나온 모델의 결과인 0.959와 비교하여 0.970이 나왔다는 점에서 너 높은 성능이 나왔다는 것을 한번 더 확인 해볼 수 있었습니다.

[Q7] 과제 2를 수행하기 위해 사용된 데이터셋(Dataset 1)과 이번 과제 수행을 위해 선택된 데이터셋 (Dataset 2)에 대해서 각각 로지스틱 회귀분석을 수행하여 Test Dataset에 대한 다음 Confusion Matrix를 채우고 이에 대한 결과를 해석해 보시오.

우선 로지스틱 회귀 모델을 돌렸을 경우 나오게 되는 summary는 위와 같다는 것을 확

인 할 수 있습니다. 이때 test와 train은 60:40으로 나누었는데 그 이유는 이전에 진행하였던 모델들과 공평한 비교를 하기 위해서 해주었습니다. 이에 따라 나오게 되는 confusion matrix와 여러 성능 평가 지표들은 다음과 같습니다.

	lr_predicted	
lr_target	0	1
0	595	32
1	14	626

<로지스틱 회귀 모델>

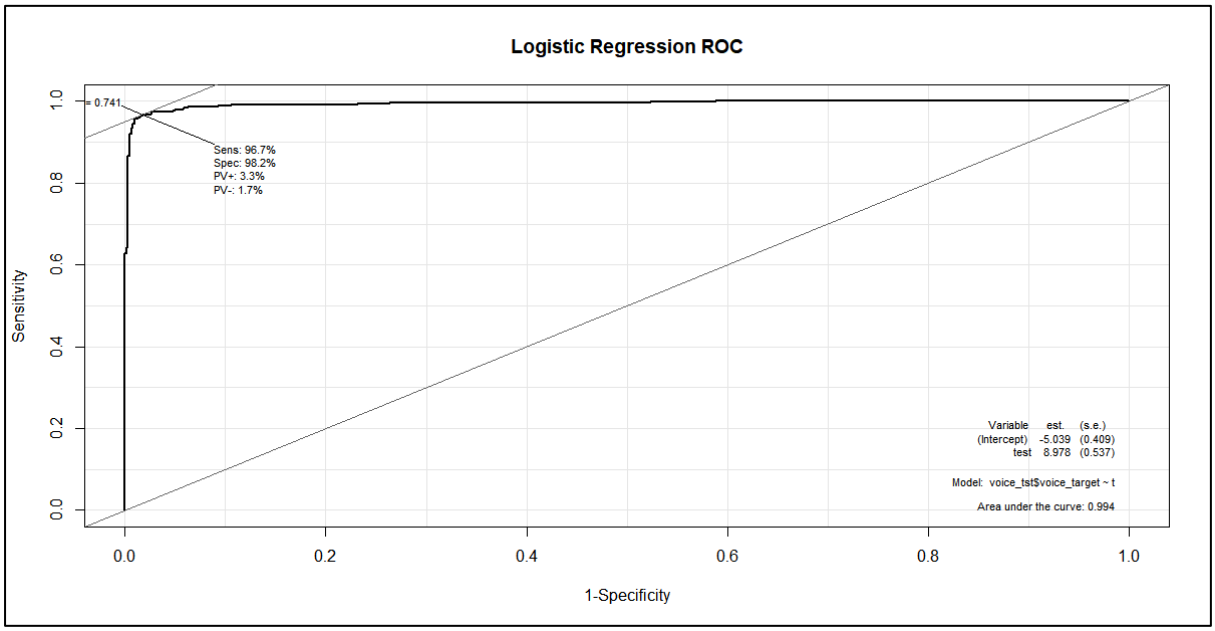
	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Logistic Regression	0.978125	0.9513678	0.9489633	0.9636938	0.9634338	0.9645609

<의사결정나무 모델>

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Non-Pruning	0.9515625	0.9666667	0.9665072	0.9589582	0.9590057	0.9590551
Post-Pruning	0.9515625	0.9666667	0.9665072	0.9589582	0.9590057	0.9590551
Pre-Pruning	0.9723127	0.9660194	0.9678407	0.9700079	0.9700741	0.9691558

이를 토대로 해석을 해보자면 총 627명의 여자 중에서 남자라고 오분류된 경우는 32건이 있었고, 640명의 남자 중에서 여자라고 오분류되는 값은 총 14건이었다는 것을 확인해 볼 수 있었습니다. 위에서 제가 결론 내린 가장 좋은 의사결정나무 모델은 pre-pruning 의사 결정 나무였는데 이와 비교해 보았을 때, 남성인데 여자라고 오분류 되는 건수는 3건이 줄었으나 여성인데 남성의 목소리라고 오분류되는 건수가 9건이 증가하였다는 것을 확인 할 수 있었습니다. 그 외의 성능 지표를 보면, 나머지 세개의 모델들과 비교해보았을 때, TPR은 가장 높았으며 Precision은 가장 작았고, 이에 따라 TNR, Accuracy는 가장 작았으며, BCR과 F1-measure은 pre-pruning 모델에 미치지 못하였다는 점을 확인 할 수 있었습니다.

추가적으로 AUROC를 확인해보면, 다음과 같은 결과를 얻을 수 있었습니다.



AUC는 0.994로 의사결정나무의 어떤 모델과 비교해도 가장 낮은 값을 지녔다는 것을 확인 할 수 있었습니다. 이에 따라 지금까지 구축한 모델들 중, 가장 성능이 떨어지는 로지스틱이 가장 성능이 떨어지는 모델이 아닐까? 라는 합리적 의문을 가졌습니다.

[Q8] 각 데이터셋마다 Logistic Regression에 의해 중요하다고 판별된 변수들과 의사결정나무에 의해 중요하다고 판별된 변수들을 확인해보고 차이가 있는 지의 여부와, 차이가 존재할 경우 그 이유에 대한 본인의 생각을 서술해 보시오

```
glm(formula = voice_target ~ ., family = binomial, data = voice_trn)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7164  -0.0393  -0.0002   0.1003   4.2955

Coefficients: (3 not defined because of singularities)
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.96140    0.23202  -4.144 3.42e-05 ***
meanfreq     0.50937    1.69709   0.300 0.764065
sd           0.22121    0.74643   0.296 0.766959
median      -0.35285    0.59370  -0.594 0.552294
Q25         -2.86278    0.70160  -4.080 4.50e-05 ***
Q75          1.28685    0.59182   2.174 0.029675 *
IQR              NA         NA      NA       NA
skew         0.82991    0.90909   0.913 0.361295
kurt        -1.10006    0.75318  -1.461 0.144137
sp.ent       2.29653    0.60419   3.801 0.000144 ***
sfm         -2.07358    0.60589  -3.422 0.000621 ***
mode         0.08112    0.22310   0.364 0.716139
centroid         NA         NA      NA       NA
meanfun     -5.38656    0.36334 -14.825  < 2e-16 ***
minfun       0.73817    0.26082   2.830 0.004652 **
maxfun      -0.06856    0.27165  -0.252 0.800737
meandom     -0.04805    0.31809  -0.151 0.879919
mindom       0.07491    0.20091   0.373 0.709254
maxdom       0.32280    0.33383   0.967 0.333572
dfrange          NA         NA      NA       NA
modindx     -0.30311    0.26345  -1.151 0.249928
```

위의 summary 표를 확인해보면 Q25, Q75, sp.ent, sfm, meanfun, minfun이 로지스틱 회귀에서 중요한 변수로 선정되었다는 것을 확인 해볼 수 있었습니다. 반면 tree에서 중요하다고 선정된 변수 들에는 meanfun, IQR, Q25, Q75, sfm, modindx, sp.ent, meandom 이 있다는 것을 확인 할 수 있었습니다. 이러한 차이가 있는 이유는 가장 먼저 구조적으로 차이가 있기 때문이라고 생각합니다. 우선 제가 느낀 결론부터 말하자면, 의사결정나무의 구조는 로지스틱에 비해 단순하다는 느낌이 들었습니다. 물론 이러한 점이 직관적인 결과 비교를 가능케 하지만 뭔가 딱 결정적인 답을 내주지는 않는 느낌입니다. 단순하다는 성질이 중요한 변수의 개수가 늘어나는 결정적인 이유인 것 같습니다. 비유를 하자면 의사결정 나무는 어떤 결과를 내리기 까지 몇 개의 가지를 쳐서 "이러한가? 이렇지 않으면 반대로 가자" 식이고 당연히 성능을 높이기 위해서는 많은 질문을 하는 것이 유리하기 때문인 것 같습니다. 하지만 로지스틱은 어떤 변수 하나하나를 보는 것이 아닌 조금 고차원적인 듯한 경향이 있어 굳이 많은 변수를 사용하지 않아도 되는 것이라고 생각합니다.

<심장병 예측 데이터>

[Q1] 본인이 생각하기에 "예측 정확도"도 중요하지만 "예측 결과물에 대한 해석"이 매우 중요할 것으로 생각되는 분류 문제를 다루고 있는 데이터셋을 1개 선정하고 선정 이유를 설명하시오. 데이터셋 탐색은 아래에서 제시된 Data Repository를 포함하여 여러 Repository를 검색해서 결정하시오. 보고서에는 데이터를 다운로드할 수 있는 링크를 반드시 제공하시오.

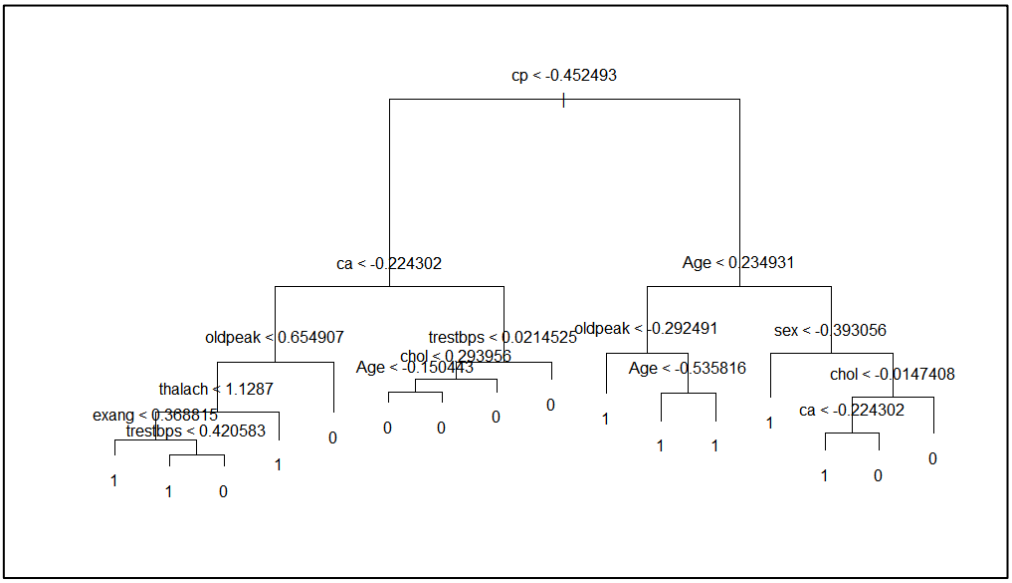
과제 2에서 사용한 데이터는 목소리의 성질을 이용하여 여성의 목소리인지 남성의 목소리인지를 구분하는 단순 분류 데이터 였습니다. 사실상 목소리를 가지고 얼마나 정확하게 성별을 가려낼 수 있는 가는 흥미로운 주제지만 중요하다고 할 만한 데이터는 아니라고 생각합니다. 하지만 두번째 파트에서 제가 사용할 데이터는 심장병과 관련된 데이터 입니다. 나이, 성별, 외 건강 정보들을 이용하여 심장병이 있는지 없는 지를 판단하기 위해 이용되는 데이터로서 어떻게 보면 매우 중요한 내용을 다루고 있다고 할 수 있을 것 같습니다. 정확도가 얼마나 나오느냐를 떠나서 예측 결과물이 어떻게 나왔고, 이에 따라 어떤 결론을 내어 당사자에게 전달을 해야 할 지가 더 중요한 데이터 셋이라고 할 수 있습니다. 좀더 팩트에 관한 정보를 덧붙이자면, 심혈관 질병은 2019년도 전세계 사망자 중 1/3이나 차지할 정도로 주요 선두 원인이므로 이에 대한 분석은 더 많이 이루어 져야 한다는 생각 또한 이 데이터셋의 선정 원인 중 하나가 될 것 같습니다.

링크: https://www.kaggle.com/ronitf/heart-disease-uci

데이터를 train과 test 로 나누는 과정에서 저는 60%는 training으로 40%는 test로서 이용하였습니다. 그 이유는 너무 많은 데이터를 training 에 이용하면 과적합이 발생할 수도 있을 것 같다는 생각을 하였기 때문이고, 굳이 반으로 나누지 않은 이유는 아무리 과적합이 발생할 것 같다 하더라도 너무 모델 학습이 잘 되지 않으면 소용이 없을 것 같다는 생각을 하였기 때문입니다.

[Q2] 실습 시간에 사용한 "tree" package 를 사용하여 Classification Tree 를 학습한 결과물을 Plotting 하고 이에 대한 해석을 수행하시오. 또한 해당 Tree 를 pruning 을 수행하지 않은 상태에서 Test dataset 에 대한 분류 성능을 평가하시오.

우선 pruning을 시행하지 않은 상태에서 데이터를 돌려 결과물을 확인 해본 결과는 다음과 같았습니다.



겹치는 글들이 많아 해석을 하는데 있어 분석을 하는데 꽤나 어려움이 느껴졌습니다. 하지만 summary를 확인함으로서 더 자세한 분석을 할 수 있었습니다.

```
> summary(CART_post)

Classification tree:
tree(formula = voice_target ~ ., data = voice_trn)
Variables actually used in tree construction:
[1] "cp"       "ca"       "oldpeak"  "thalach"  "exang"    "trestbps"
 "chol"
[8] "Age"      "sex"
Number of terminal nodes:  16
Residual mean deviance:  0.4695 = 77.94 / 166
Misclassification error rate: 0.1044 = 19 / 182
```

위의 summary를 확인 해보면 총 9개의 변수가 분할 변수로서 사용이 되었다는 점을 확인 할 수 있었습니다.

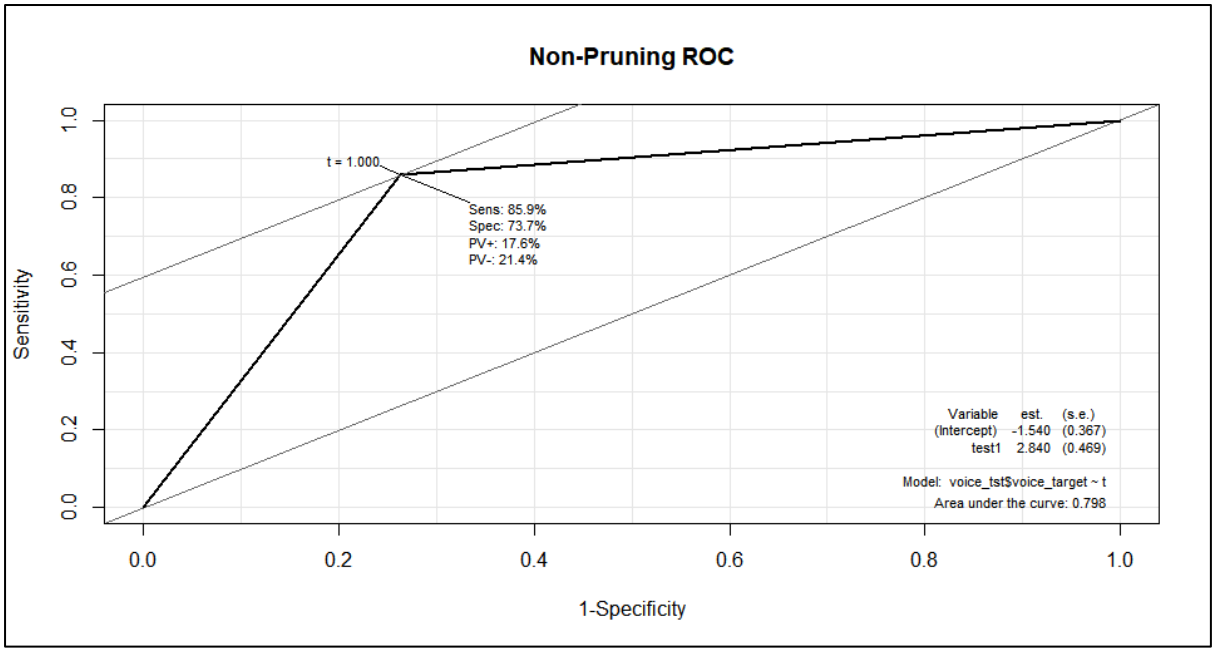
또한, 혼동 행렬과 성능 평가 지표들을 분석해보았을 때의 결과물은 다음과 같았습니다.

```
  CART_post_pray
    0  1
0 42 15
1  9 55
```

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Non-Pruning	0.859375	0.7857143	0.7368421	0.8016529	0.7957535	0.8208955

우선 혼동 행렬에 대해 살펴보자면, 심장병이 없는데 있다고 판단한 결과는 57명 중 15명이였고, 심장병이 있는데 없다고 판단이 내려진 사람들은 64명 중 9명이였습니다. 이에 따른 TPR, Precision, TNR, Accuracy, BCR, F1measure은 위와 같이 나왔습니다.

AUROC의 경우는 다음과 같았습니다.

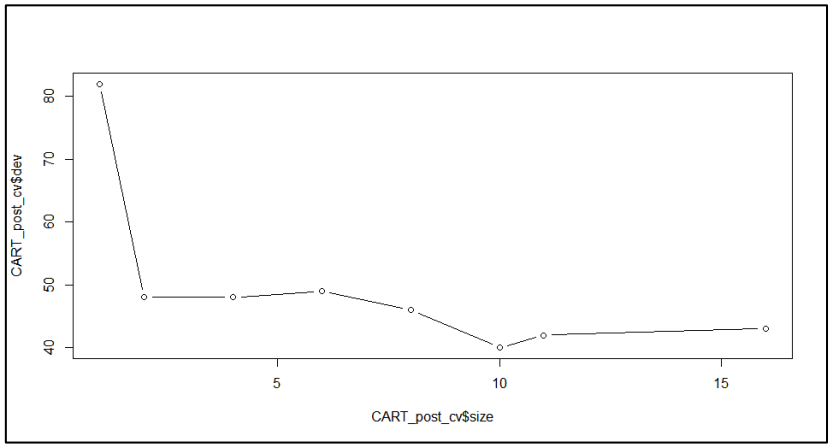


Area under the curve 는 0.798이라는 수치가 나왔다는 것을 확인 할 수 있었습니다.

하지만 이 데이터셋의 경우 제가 개인적으로 더 나은 성능을 가지고 있다고 판단할 수 있는 지표는 환자가 심장병이 있는 없다고 판단을 내리는 횟수를 낮은 것이라고 생각합니다. 그만큼 심장병이 있는 사람한테 없다고 진단하면 그 사람은 정말 죽을 수도 있기 때문입니다. 앞으로 나올 다름 모델들과는 이를 중점으로 비교를 해보고자 합니다.

[Q3] 앞에서 생성한 Tree에 대해서 적절한 Post-Pruning을 수행한 뒤 결과물을 Plotting하고 이에 대한 해석을 수행하시오. Pruning 전과 후에 Split에 사용된 변수는 어떤 변화가 있는가? Test dataset에 대한 분류 성능을 평가하고 [Q2]의 결과와 비교해보시오.

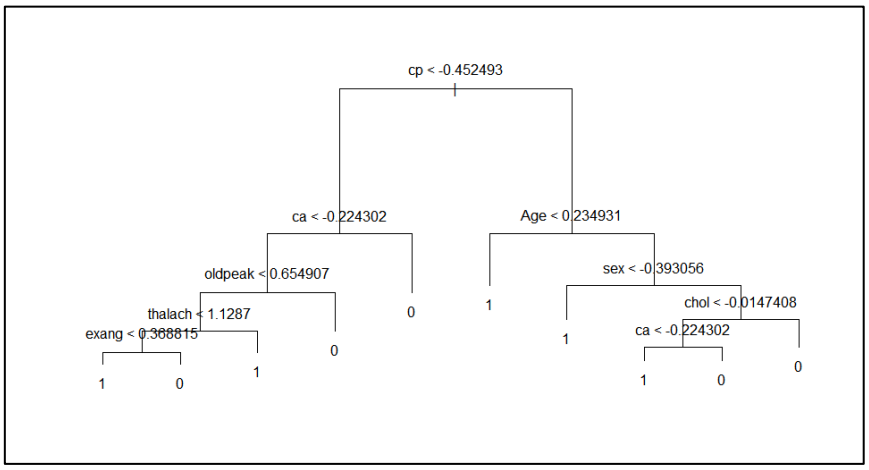
다음으로는 post pruning을 수행하여 tree 모델을 구축해보도록 하겠습니다. 현재 제가 돌린 트리 모형을 가지고 최적의 모델을 찾아보도록 하겠습니다.



y축은 불순도이므로 처음 스플릿이 되자마자 급격하게 떨어지는 것을 확인할 수 있었고, 그후로 큰 변화를 보이지 않고 있다가 사이즈가 10이 되었을 때 가장 낮은 수치를 보여준다는 것을 확인 할 수 있었습니다. 오른쪽 표를 통해 위 그래프에 대한 더 상세한 정보를 확인 할 수 있었습니다. 사이즈가 1일때는 82였으나 이후 사이즈가 2가 되었을 때 급격하게 48로 줄어든 것을 확인 하였고, 이후 미세한 변동만 있다가 10에서 가장 낮은 불순도를 보여주고 다시 아주 미세하게 상승한다는 것을 확인 할 수 있었습니다. 이를 통해 현재 10개의 리프노드를 가지고 있을 때 불순도는 40으로 가장 낮으니까 최적의 모델이겠구나 라고 생각하였습니다.

```
$size
[1] 16 11 10  8  6  4  2  1

$dev
[1] 43 42 40 46 49 48 48 82

$k
[1] -Inf  0.0  1.0  2.0  2.5  3.0  4.5 37.0

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

사이즈를 10으로 하였을 때, 나무 모형을 plotting 해본 결과는 아래에 있는 그림과 같이 나오는 것을 확인 할 수 있었습니다.



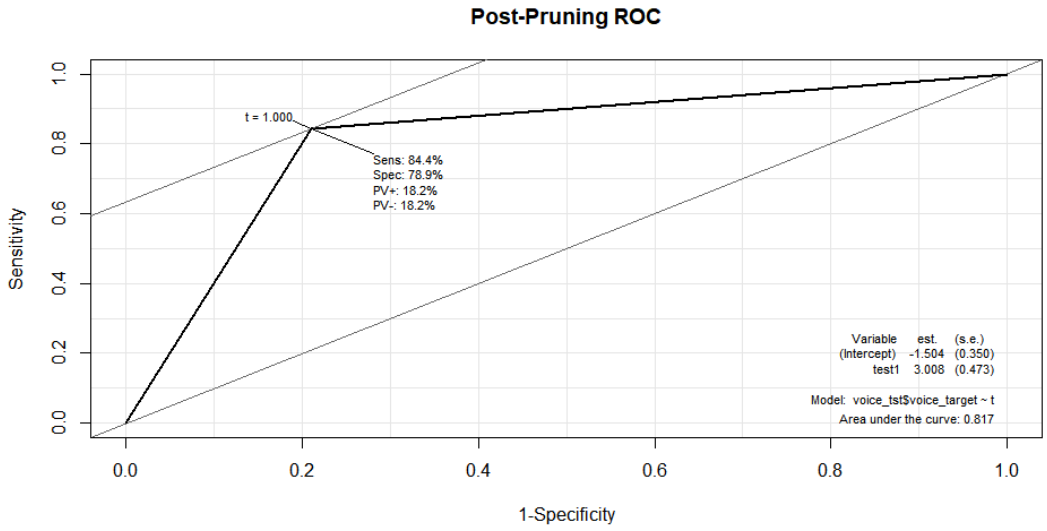
규칙들에 대해서는 다음과 같이 형성이 되어있다는 것을 확인 할 수 있었고, 성능을 평가하기 위해 혼동행렬과 여러 평가 지표들을 계산해본 결과는 아래와 같았습니다.

	CART_post_prey	
	0	1
0	45	12
1	10	54

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Non-Pruning	0.859375	0.7857143	0.7368421	0.8016529	0.7957535	0.8208955
Post-Pruning	0.843750	0.8181818	0.7894737	0.8181818	0.8161608	0.8307692

혼동 행렬을 보자면 심장병이 있는 총 64명의 사람 중 10명이 없다고 오 분류 되어 non-pruning 일때보다 한명 더 늘어났다는 것을 확인 할 수 있었습니다. 반면 심장병이 없는 사람 총 57명 중 12명이 오분류 되었는데, 결과적으로 바라본다면 TPR 을 제외한 모든 수치가 증가하여 더 나

은 모델이라고 보이나, 위에서 언급한 바와 같이 심장병이 있는데 없다고 판단되는 사람의 수가 더 적어야 한다고 생각하는 입장이므로 더 좋은 모델인지에 대해서는 의문이 들었습니다. 추가적으로 AUROC는 다음과 같이 나왔습니다.



AUC는 이전 결과인 0.798보다 더 높은 0.817이 나왔다는 점에서 성능이 더 좋아진 것은 맞다고 생각합니다. 하지만 거듭 강조하지만 성능이 더 좋아졌다고 해서 이 데이터셋에서 더 좋은 모델이라고는 생각하지 않습니다.

[Q4] "party" package를 사용하여 다음 조건에 맞는 Pre-pruning을 수행하여 가장 최적의 min_criterion, min_split, max_depth 값을 찾아보시오.

우선, pre-pruning 을 하기 전 이번에는 60%의 데이터를 이전 non과 post-pruning 과정을 거쳤을 때와 동일한 비율로 training을 시키기 위해, 먼저 trainset은 전체 데이터의 50%로 생성하였고, 나머지 50%의 데이터 중, 20%는 검증용 데이터, 나머지는 test용 데이터로 하였습니다. 결론적으로는 공평한 비교를 위해 이전 모델에서 분리한 train을 위한 데이터와 test를 위한 데이터의 비율이 같아지게 하였습니다. 단순히 60%를 training 으로 이용하고자 한 이유는 너무 많은 데이터의 수를 training으로 하면 과적합이 발생할 것 같다는 생각이 들어서 였으며, 이전 모델과 새로운 모델을 비교할 때, 동일한 비율로 train과 test를 나누는 것이 모델의 성능을 파악하기 더 용이 할 거라고 생각하여 위와 같이 데이터를 나누게 되었습니다.

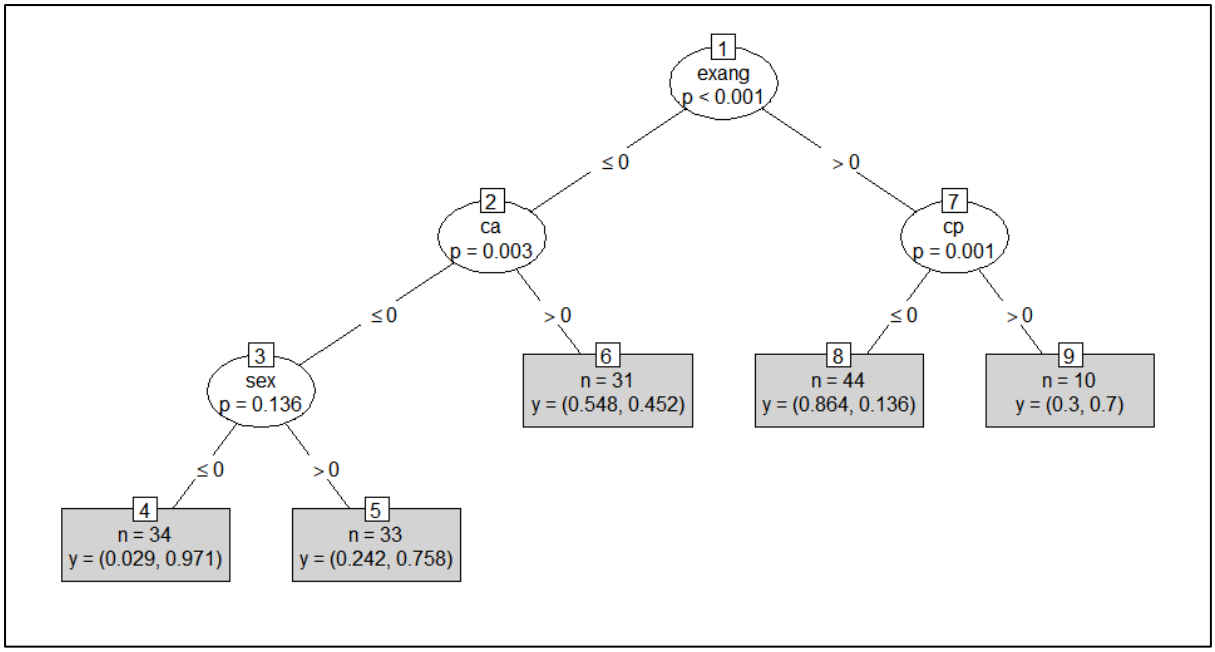
첫번째 파트에서 후보군으로 올린 값과 동일한 값을 넣어 iteration을 돌려 보았습니다. 이 과정을 통해 AUROC를 기준으로 선정된 최적의 하이퍼 파라미터 들은 다음과 같았습니다.

	min_criterion	min_split	max_depth	TPR	Precision	TNR	ACC	BCR	F1
[1,]	0.50	50	0	0.7647059	0.8666667	0.8461538	0.8000000	0.8043997	0.8125000
[2,]	0.50	50	5	0.7647059	0.8666667	0.8461538	0.8000000	0.8043997	0.8125000
[3,]	0.50	50	10	0.7647059	0.8666667	0.8461538	0.8000000	0.8043997	0.8125000
[4,]	0.50	50	15	0.7647059	0.8666667	0.8461538	0.8000000	0.8043997	0.8125000
[5,]	0.50	50	20	0.7647059	0.8666667	0.8461538	0.8000000	0.8043997	0.8125000
[6,]	0.60	50	0	0.7647059	0.8666667	0.8461538	0.8000000	0.8043997	0.8125000
[7,]	0.60	50	5	0.7647059	0.8666667	0.8461538	0.8000000	0.8043997	0.8125000
[8,]	0.60	50	10	0.7647059	0.8666667	0.8461538	0.8000000	0.8043997	0.8125000
[9,]	0.60	50	15	0.7647059	0.8666667	0.8461538	0.8000000	0.8043997	0.8125000

AUROC	N_leaves
0.8371041	5
0.8371041	5
0.8371041	5
0.8371041	5
0.8371041	5
0.8371041	5
0.8371041	5
0.8371041	5
0.8371041	5
0.8371041	5

Min_criterion은 0.5, min_split은 50, max_depth는 0이라는 결과를 얻을 수 있었습니다. 이때의 AUROC 값은 0.8371041의 값을 가지고 있다는 것을 확인 할 수 있었습니다. 선정된 하이퍼 파라미터를 이용하여 구축한 모델에 대한 분석은 아래에서 마저 진행하도록 하겠습니다.

[Q5] 최적의 결정나무의 Plot을 그리고, 대표적인 세 가지 규칙에 대해서 설명해보시오.



우선 가장 처음으로 사용된 분할 변수는 exang으로 협심증(통)의 여부를 나타내는 변수입니다. 만약 협심통이 있다면 chest pain type를 보게 됩니다. 위의 자료를 토대로 본다면 chest pain의 타입은 데이터 상 총 4가지가 있지만 0번째를 제외한 나머지 타입들에 대해서는 심장병일 확률이 0.138이지만 만약 타입이 0이라면 70%의 확률로 심장병이라고 합니다. 그리고 이 조건을 확인하는 데 사용된 총 데이터 수는 10명이라는 것도 확인 할 수 있었습니다. 또한 나무의 왼쪽

으로 가보자면 성별에 따라 exang가 0이고, ca(number of major vessels)가 0보다 작은 데이터 중 남성 (sex=1)인 사람중 75.8%는 심장병이라는 결과가 나왔다는 것을 확인 할 수 있었습니다.

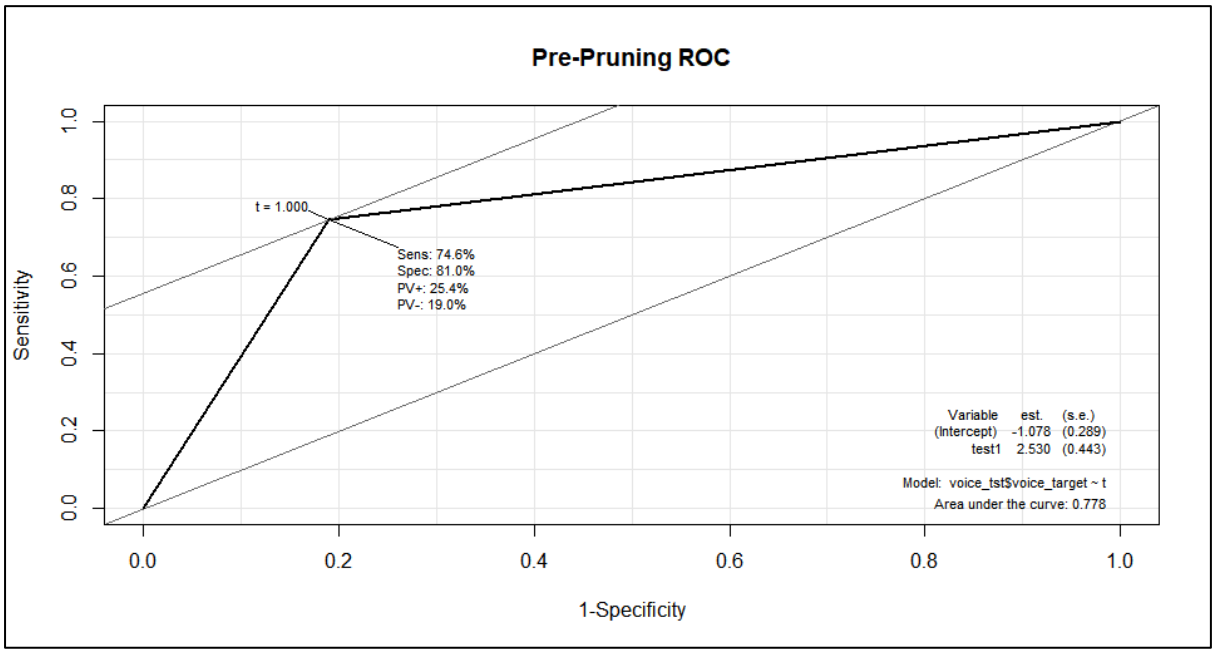
[Q6] [Q4]에서 선택한 하이퍼라라미터 조합을 이용하여 Training Dataset과 Validation Dataset을 결합한 데이터셋을 학습한 뒤, Test Dataset에 적용해보고 분류 성능을 평가하시오.

이 또한 이전과 마찬가지의 순서로 비교를 해보고자 합니다. 혼동행렬과 평가 지표들을 보면 다음과 같습니다.

```
 CART_pre_prediction
   0  1
0 47 11
1 16 47
```
15,9

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Non-Pruning	0.8593750	0.7857143	0.7368421	0.8016529	0.7957535	0.8208955
Post-Pruning	0.8437500	0.8181818	0.7894737	0.8181818	0.8161608	0.8307692
Pre-Pruning	0.7460317	0.8103448	0.8103448	0.7768595	0.7775236	0.7768595

혼동 행렬을 보면, Non과 Post와 비교해보았을 때, 심장병이 있는 사람을 없다고 판단한 비율이 가장 높았습니다. 또한 심장병이 없는데 있다고 오분류한 경우의 수도 11로 가장 많았던 post의 12라는 수치보다 하나 작았다는 것을 확인 할 수 있었습니다. 이에 따라 여러 방면으로 성능이 떨어졌다는 것을 확인해 볼 수 있었습니다



AUROC를 확인해보면 0.778로 가장 낮은 수치가 나왔다는 것을 확인해 볼 수 있었습니다.

[Q7] 과제 2를 수행하기 위해 사용된 데이터셋(Dataset 1)과 이번 과제 수행을 위해 선택된 데이터셋 (Dataset 2)에 대해서 각각 로지스틱 회귀분석을 수행하여 Test Dataset에 대한 다음 Confusion Matrix를 채우고 이에 대한 결과를 해석해 보시오.

```
Call:
glm(formula = heart_target ~ ., family = binomial, data = heart_trn)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2899  -0.3031   0.1226   0.5433   2.6210

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.01462    0.26020   0.056 0.955180
Age         -0.24500    0.32224  -0.760 0.447081
sex         -0.93132    0.30138  -3.090 0.002000 **
cp           0.90412    0.26870   3.365 0.000766 ***
trestbps    -0.55093    0.26127  -2.109 0.034974 *
chol        -0.11409    0.25568  -0.446 0.655426
fbs          0.02204    0.26069   0.085 0.932633
restecg      0.38463    0.26447   1.454 0.145844
thalach      0.58892    0.35286   1.669 0.095117 .
exang       -0.54157    0.25585  -2.117 0.034282 *
oldpeak     -1.02234    0.34882  -2.931 0.003380 **
slope        0.11766    0.29677   0.396 0.691746
ca          -0.74031    0.26067  -2.840 0.004511 **
thal        -0.56407    0.25987  -2.171 0.029966 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)
```

우선 로지스틱 회귀 모델을 돌렸을 경우 나오게 되는 summary는 위와 같다는 것을 확인 할 수 있습니다. 이때 test와 train은 60:40으로 나누었는데 그 이유는 이전에 진행하였던 모델들과 공평한 비교를 하기 위해서 해주었습니다. 이에 따라 나오게 되는 confusion matrix와 여러 성능 평가 지표들은 다음과 같습니다.

	lr_predicted	
lr_target	0	1
0	47	10
1	9	55

<로지스틱 회귀 모델>

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Logistic Regression	0.859375	0.8461538	0.8245614	0.8429752	0.8417882	0.8527132

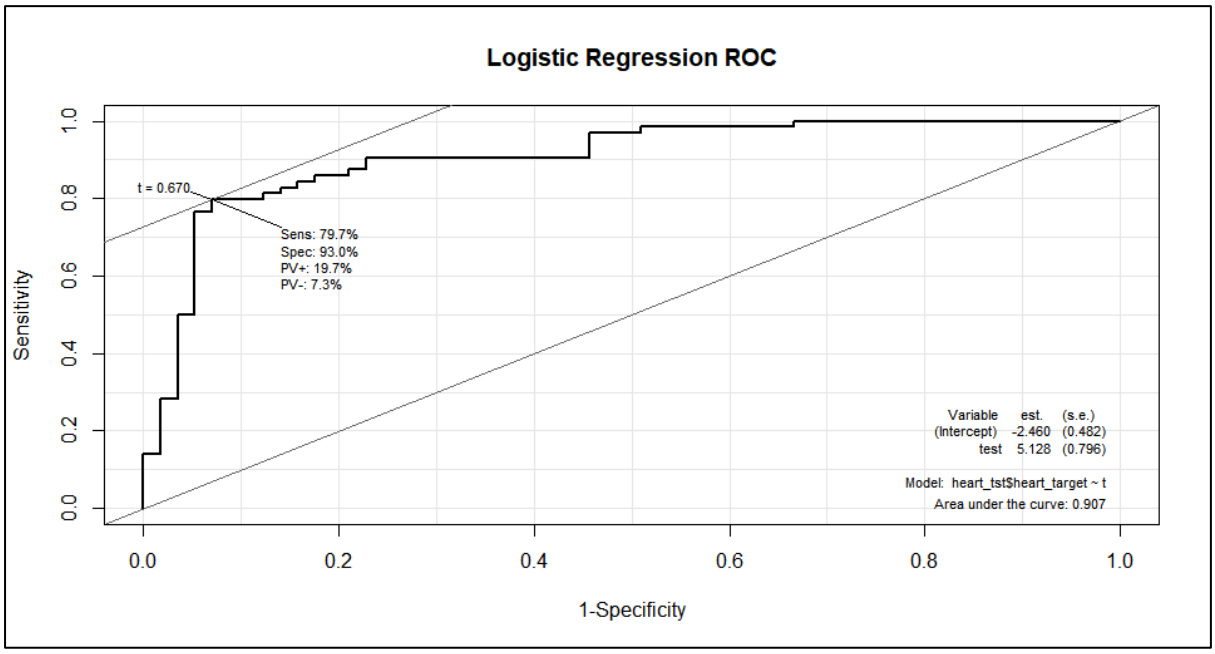
<의사결정나무 모델>

	TPR	Precision	TNR	Accuracy	BCR	F1-Measure
Non-Pruning	0.8593750	0.7857143	0.7368421	0.8016529	0.7957535	0.8208955
Post-Pruning	0.8437500	0.8181818	0.7894737	0.8181818	0.8161608	0.8307692
Pre-Pruning	0.7460317	0.8103448	0.8103448	0.7768595	0.7775236	0.7768595

혼동행렬을 토대로 보았을 때, 로지스틱의 결과를 확인해보면 57명의 정상인 중 10명을 오분류 하였고, 64명의 심장병 보유자 중 9명이 정상이라고 오분류 되었다는 것을 확인 할 수 있었습니다. 이는 오분류 율이 위의 decision tree 모델들 중 가장 낮은 수치라는 것을 확인 할 수 있었습니다. 또한 이에 따라서 TPR은 Non pruning decision tree 모델과 같은 수치라는 것을 확인

할 수 있었고, 나머지 값들은 어느 decision tree 모델들 보다 더 좋은 성능을 보여주었습니다.

이에 따라 AUROC의 값도 가장 높을 것이라고 예상하였는데, 결과는 다음과 같았습니다.



AUC의 값이 역시나 0.907으로써 압도적으로 높은 값을 보여주고 있다는 것을 확인 할 수 있었습니다.

[Q8] 각 데이터셋마다 Logistic Regression에 의해 중요하다고 판별된 변수들과 의사결정나무에 의해 중요하다고 판별된 변수들을 확인해보고 차이가 있는지의 여부와, 차이가 존재할 경우 그 이유에 대한 본인의 생각을 서술해 보시오

로지스틱 회귀 모델에서 pvalue가 0.05보다 낮은 변수로는 sex, cp, trestbps, exang, oldpeak, ca, thal 이라는 것을 확인 할 수 있었습니다. 반면, 의사결정 나무에서 사용된 변수들로는 가장 높은 성능을 보여주었다고 판단되는 non-pruning 의사결정나무를 기준으로, cp, ca, age, oldpeak, trestbps, thalach, chol, exang이 사용되었다는 것을 확인할 수 있었습니다. 이 또한 이전 데이터셋에서 얻을 수 있었던 결론과 비슷하게 의사결정 나무에서 더 많은 변수가 사용되었음을 확인 할 수 있었습니다. 다시 한번 정리하자면, 로지스틱 회귀 모델은 의사결정나무 모델보다 더 함축적인 의미를 담아서 변수를 사용하는 것 같습니다. 즉 하나의 변수를 바라볼 때 다른 변수간의 상관관계를 더 깊게 따지기 때문에 적은 변수로 더 고차원적인 결론을 낼 수 있는 것 같습니다. 반면에 의사결정나무는 더 단순한 구조를 가지고 있으며, 이는 곧 고 성능을 얻기 위해서는 더 많고 세세한 질문을 통해 정확도를 높여나가야 하기에 많은 변수를 사용하는 것 같다는 생각해볼 수 있었던 것 같습니다.