채널 G
2016125077 최재혁
2016126049 박희재
2016125069 조세희

데이터 사이언스

- 6주차 팀 과제 (의사결정트리)

목 차

- 과제 설명

  결정나무의 가지치기.

- 과제 목표

  사전가지치기와 사후가지치기를 적용하여 결정나무를 생성하고 두 가지치기 방법에 대하여 성능 비교를 한다.

- 사용 Tool

  분석용 언어 : R

목 차

A~J까지 순서대로 설명할 예정입니다

- 과제 1,2
- 과제 3,4
- 과제 5,6
- 과제 7
- 고찰

과제 1,2

- 과제 1 – 데이터에 대하여 간단한 데이터 탐색과정 – EDA 를 실행하라

  - Titanic Dataset 사용 : 사전에 data탐색을 통하여 목표변수와 종속변수 간 연관성 찾아보았다.
  - 밑의 자료들은 csv 파일 내에서 pivot을 하여 시각화 한 자료이다.

- 도출한 연관성

  - 사망한 인원(809명)이 생존한 인원(500명)보다 약 1.5배 많다.
  - 주로 남성인 경우에 사망하는 경우가 많았다.
  - 1등석에 탄사람의 생존율이 가장 높고, 3등석이 가장 낮다.

sex	남	여
Survived	161	339
death	682	127

□pclass	1□	2□	3□
Survived	200	119	181
death	123	158	528

age	Death	Survived
Na	190	73
0-10	32	50
11-20	83	56
21-30	202	119
31-40	109	83
41-50	71	46
51-60	29	26
61-70	19	6
71-80	4	2

과제 1,2

- 과제 2 – 결측치(missing data)에 대하여 조사, 결측치를 적절히 채워 넣을 것
  - 결측치 존재여부 파악 → 종속변수 중 결측치 존재
  - Na.omit() 함수를 사용, 결측치 제거 → 1310명 1045명의 data 추출

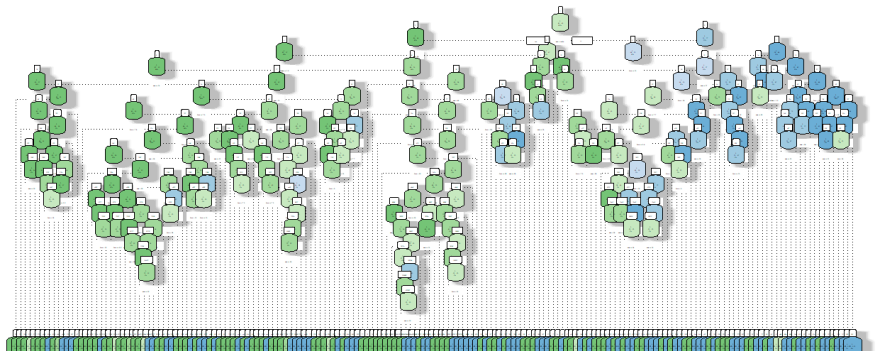
Full-Tree 생성

- Full-Tree 생성 – 가치치기들을 수행하기 전 full-tree를 생성한다

  - <사진1>을 통해 full tree에서 잎 노드가 무수히 많음을 알 수 있다. 그렇기에 과적합의 위험이 발생함으로 가지치기가 필요함.

  - <사진2>는 테스트셋과 비교, 성능평가 confusionMatrix() 사용
    Accuracy : 0.7284, Sensitivity : 0.7747, Specificity : 0.6641

  - 원본데이터에서 Death(0)[809], survived(1)[500] Death에 편중
    createDataPartition()사용 dataset의 generalization x



<사진 1> full-tree 모습

```
> #일단 full tree에서의 정확도를 측정
> rpartpred<-predict(fit, test, type='class')
> rpartpred <- as.factor(rpartpred)
> confusionMatrix(rpartpred,survivpred)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 141  44
         1  41  87

               Accuracy : 0.7284
                 95% CI : (0.6756, 0.7769)
    No Information Rate : 0.5815
    P-Value [Acc > NIR] : 4.642e-08

                  Kappa : 0.4403

 Mcnemar's Test P-Value : 0.8283

            Sensitivity : 0.7747
            Specificity : 0.6641
         Pos Pred Value : 0.7622
         Neg Pred Value : 0.6797
             Prevalence : 0.5815
         Detection Rate : 0.4505
   Detection Prevalence : 0.5911
      Balanced Accuracy : 0.7194

       'Positive' Class : 0
```

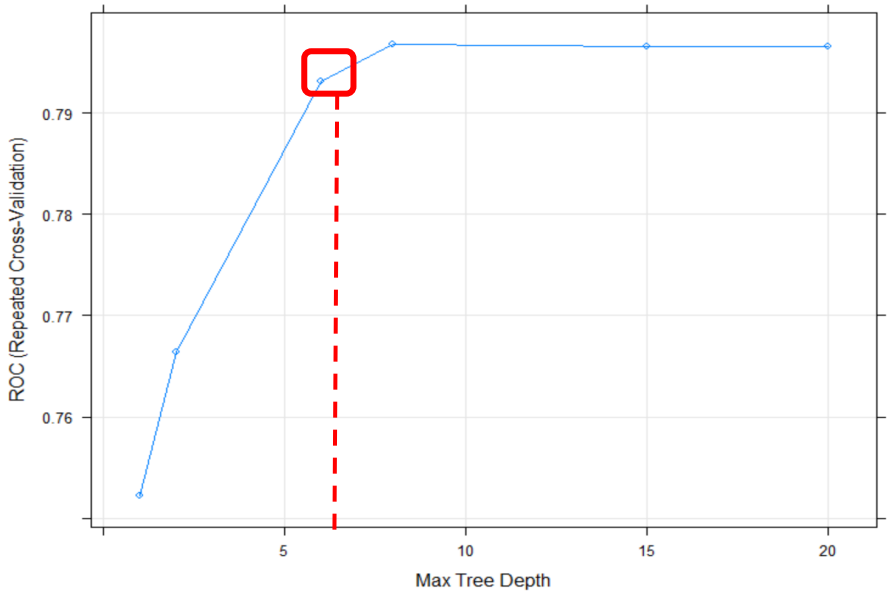
<사진 2> full-tree 성능평가

과제 3

- 과제 3 – 사전 가치치기 Maxtree depth를 최적화

```
// 사전 가지치기 모델의 depth 최적화를 위한 모델 깊이에 따른 정확도 시각화

system.time (postprun <- train(survived~., data=train,
                               method = "rpart2",
                               tuneLength = 8,
                               trControl = ctrl,
                               metric = "ROC"))
plot(postprun)
```

<사진 1> depth 최적화를 위한 depth 에 따른 정확도 시각화



<사진 2> 시각화 된 graph

- 시각화 된 그래프에서 ROC값의 변화가 멈추기 직전 고점일 때의 depth 대입
  MAX_tree depth 최적화 -> depth 이용 사전 가지치기

과제 3

- 과제 3 – 사전 가치치기 Maxtree depth를 최적화

- <사진1> 사전가치지기 모델의 성능평가
  <사진2> 가지치기 하지 않은 모델의 성능평가.

	사전가지치기 모델	FULL TREE 모델
Accuracy	0.8115	0.7284
Sensivility	0.9121	0.7747
specificity	0.6718	0.6641

```
> rpartpred4<-predict(rtree_model, test, type='class')
> rpartpred4 <- as.factor(rpartpred4)
> confusionMatrix(rpartpred4,survivpred)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 166  43
         1  16  88

               Accuracy : 0.8115
                 95% CI : (0.7637, 0.8533)
    No Information Rate : 0.5815
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6012

 Mcnemar's Test P-Value : 0.000712

            Sensitivity : 0.9121
            Specificity : 0.6718
         Pos Pred Value : 0.7943
         Neg Pred Value : 0.8462
             Prevalence : 0.5815
         Detection Rate : 0.5304
   Detection Prevalence : 0.6677
      Balanced Accuracy : 0.7919

       'Positive' Class : 0
```

<사진 1> - 사전가지치기 모델 성능평가

```
> #일단 full tree에서의 정확도를 측정
> rpartpred<-predict(fit, test, type='class')
> rpartpred <- as.factor(rpartpred)
> confusionMatrix(rpartpred,survivpred)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 141  44
         1  41  87

               Accuracy : 0.7284
                 95% CI : (0.6756, 0.7769)
    No Information Rate : 0.5815
    P-Value [Acc > NIR] : 4.642e-08

                  Kappa : 0.4403

 Mcnemar's Test P-Value : 0.8283

            Sensitivity : 0.7747
            Specificity : 0.6641
         Pos Pred Value : 0.7622
         Neg Pred Value : 0.6797
             Prevalence : 0.5815
         Detection Rate : 0.4505
   Detection Prevalence : 0.5911
      Balanced Accuracy : 0.7194

       'Positive' Class : 0
```

<사진 2> - full-tree 모델 성능평가

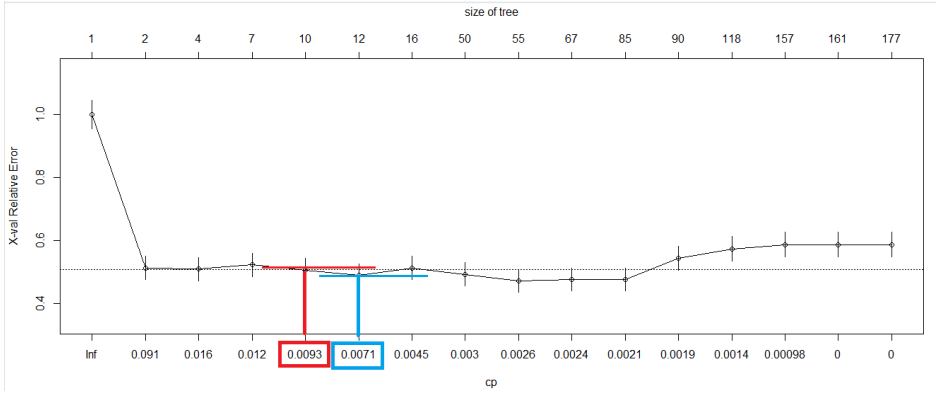
- 정확도의 모든 면에서 사전가지치기 모델의 성능이 더 우수함.
- 따라서, pruning 작업 시 성능 향상

과제 4

- 과제 4 – 사후 가지치기 - Post-pruning- cp를 최적화

- 최적화된 CP를 찾기 위해 plotcp() 함수 사용
  <사진 1> 그래프 생성

- X-val Relative Error값이 가장 작은 0.0071이 최적화 된 CP값이 되어야 하지만, SE 범위 내에서 가장 작은 모델(cp = 0.0093)을 선택, 성능평가 결과, CP값이 0.0071보다 0.0093의 결과가 더 성능이 좋게 나타나므로 0.0093을 최적화된 CP값으로 설정.



<사진 1> - Rstudio 함수

```
> #cp 중에 최적화된 값을 찾음
> fit_prune1=prune(fit,cp=0.0093)
> #cp 최적화 (사후가지치기)로 정확도 향상
> rpartpred2<-predict(fit_prune1, test, type='class')
> rpartpred2 <- as.factor(rpartpred2)
> confusionMatrix(rpartpred2,survivpred)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 168  45
         1  14  86

               Accuracy : 0.8115
                 95% CI : (0.7637, 0.8533)
    No Information Rate : 0.5815
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5994

 Mcnemar's Test P-Value : 9.397e-05

            Sensitivity : 0.9231
            Specificity : 0.6565
         Pos Pred Value : 0.7887
         Neg Pred Value : 0.8600
             Prevalence : 0.5815
         Detection Rate : 0.5367
   Detection Prevalence : 0.6805
      Balanced Accuracy : 0.7898

       'Positive' Class : 0
```

<사진 2> - 사후가지치기 모델 성능평가

과제 4

- 과제 4 – 사후 가지치기 - Post-pruning- cp를 최적화

- <사진1>는 사후가치지기 모델 성능평가
  <사진2>는 full-tree 모델 성능평가

사후 가지치기 모델의 신뢰도가 0.8115 로 0.7284인 가지치기하지 않은 모델(full-tree)보다 성능이 뛰어난 것을 확연히 보여주며, 그 외에도 sensivility가 0.9213으로 더 높아 Death 에 대한 예측이 더 뛰어날 것으로 예상되며 반대로 specificity가 0.6565로 full-tree보다(0.6641) 근소하게 낮기에 survived에 대한 예측을 full-tree에 비해 근소하게 잘 못할 것으로 예상 된다.

```
> #cp 중에 최적화된 값을 찾음
> fit_prune1=prune(fit,cp=0.0093)
> #cp 최적화 (사후가지치기)로 정확도 향상
> rpartpred2<-predict(fit_prune1, test, type='class')
> rpartpred2 <- as.factor(rpartpred2)
> confusionMatrix(rpartpred2,survivpred)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 168  45
         1  14  86

               Accuracy : 0.8115
                 95% CI : (0.7637, 0.8533)
    No Information Rate : 0.5815
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5994

 Mcnemar's Test P-Value : 9.397e-05

            Sensitivity : 0.9231
            Specificity : 0.6565
         Pos Pred Value : 0.7887
         Neg Pred Value : 0.8600
             Prevalence : 0.5815
         Detection Rate : 0.5367
   Detection Prevalence : 0.6805
      Balanced Accuracy : 0.7898

       'Positive' Class : 0
```

<사진 1> - 사후가지치기 모델 성능평가

```
> #일단 full tree에서의 정확도를 측정
> rpartpred<-predict(fit, test, type='class')
> rpartpred <- as.factor(rpartpred)
> confusionMatrix(rpartpred,survivpred)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 141  44
         1  41  87

               Accuracy : 0.7284
                 95% CI : (0.6756, 0.7769)
    No Information Rate : 0.5815
    P-Value [Acc > NIR] : 4.642e-08

                  Kappa : 0.4403

 Mcnemar's Test P-Value : 0.8283

            Sensitivity : 0.7747
            Specificity : 0.6641
         Pos Pred Value : 0.7622
         Neg Pred Value : 0.6797
             Prevalence : 0.5815
         Detection Rate : 0.4505
   Detection Prevalence : 0.5911
      Balanced Accuracy : 0.7194

       'Positive' Class : 0
```

<사진 2> - 가지치기X 모델(full-tree) 성능평가

과제 5

- 과제 5 – 위에서 생성된 두 예측 모델(사전, 사후 가지치기)의 성능을 비교하라.(정확도,민감도,특이도)

```
> rpartpred4<-predict(rtree_model, test, type='class')
> rpartpred4 <- as.factor(rpartpred4)
> confusionMatrix(rpartpred4,survivpred)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 166  43
         1  16  88

               Accuracy : 0.8115
                 95% CI : (0.7637, 0.8533)
    No Information Rate : 0.5815
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6012

 Mcnemar's Test P-Value : 0.000712

            Sensitivity : 0.9121
            Specificity : 0.6718
         Pos Pred Value : 0.7943
         Neg Pred Value : 0.8462
             Prevalence : 0.5815
         Detection Rate : 0.5304
   Detection Prevalence : 0.6677
      Balanced Accuracy : 0.7919

       'Positive' Class : 0
```

<사진 1>
사전가지치기 성능평가

```
> #cp 중에 최적화된 값을 찾음
> fit_prune1=prune(fit,cp=0.0093)
> #cp 최적화 (사후가지치기)로 정확도 향상
> rpartpred2<-predict(fit_prune1, test, type='class')
> rpartpred2 <- as.factor(rpartpred2)
> confusionMatrix(rpartpred2,survivpred)
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 168  45
         1  14  86

               Accuracy : 0.8115
                 95% CI : (0.7637, 0.8533)
    No Information Rate : 0.5815
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5994

 Mcnemar's Test P-Value : 9.397e-05

            Sensitivity : 0.9231
            Specificity : 0.6565
         Pos Pred Value : 0.7887
         Neg Pred Value : 0.8600
             Prevalence : 0.5815
         Detection Rate : 0.5367
   Detection Prevalence : 0.6805
      Balanced Accuracy : 0.7898

       'Positive' Class : 0
```

<사진 2>
사후가지치기 성능평가

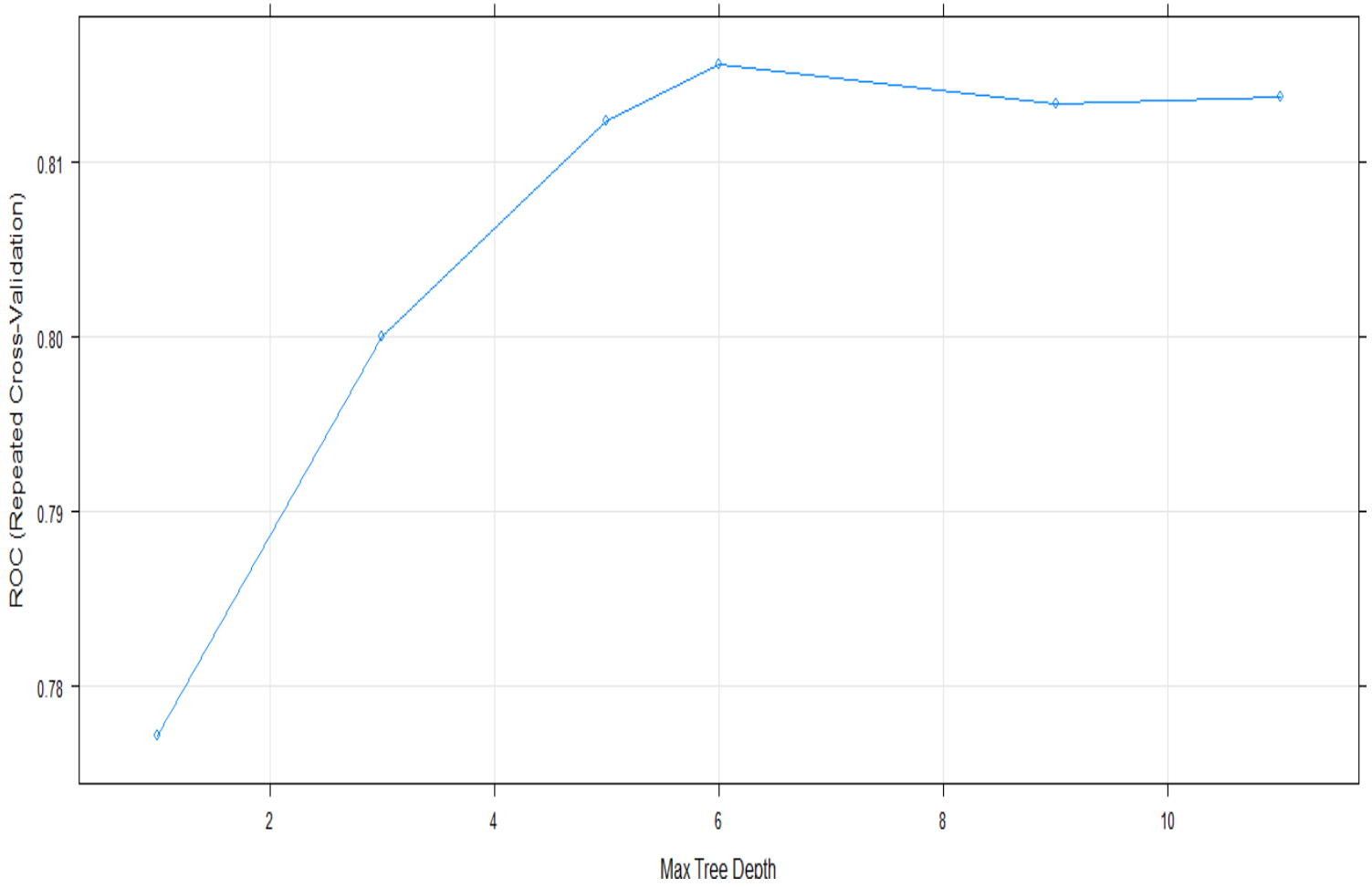
- <사진1>,<사진2> 밑줄 참고

	사전가지치기 모델	사후가지치기 모델
Accuracy	0.8115	0.8115
Sensivility	0.9121	0.9231
specificity	0.6718	0.6565

결과 : 사전가지치기 모델의 성능이 더 우수하다고 판단

과제 6

- 과제 6 – Plot 등을 이용하여 모델이 최적화되었음을 보일 것.



- 왼 쪽 사진은 사전 가지치기모델을 Plot사용 graph 시각화

- X축 Max Tree Depth
- Y축 ROC

- Max Tree Depth = 6일 때, ROC값이 최대값
  따라서, 최적화 되었음 증명

과제 7

- 과제 7 – A 위의 두 모델 중 하나로부터 rule set 을 추출 할 것.
  - rpart.rules(rtree_model)함수를 이용해 사전가지치기 한 rtree_model에서 rule set 추출. 의사결정트리에서 추출되었기에 완결적이고 상호배타적이다.
  - 위 함수를 통해 추출된 Rule set은 <사진 1>을 참고바람.

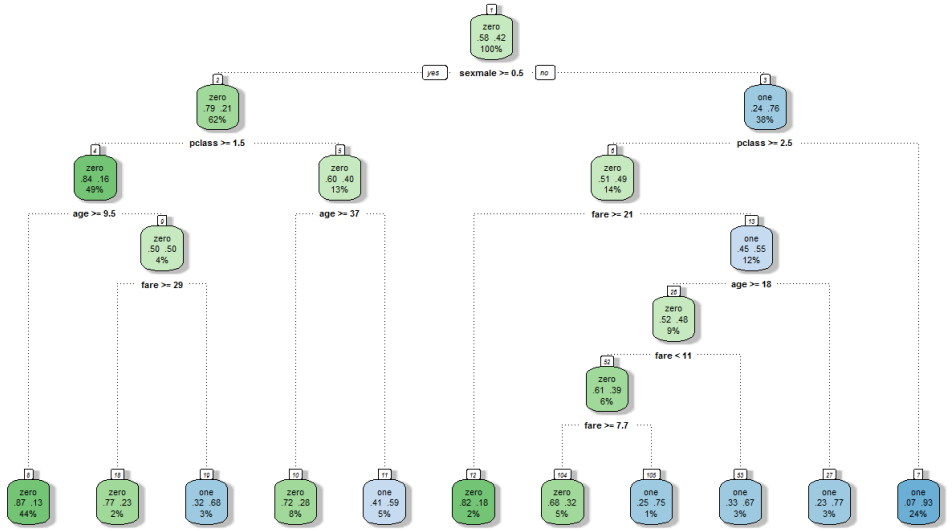
```
> rtree_model <- rpart(survived~pclass+age+sex+fare, data=train, control=rpart.control(maxdepth=6))
> rtree_model
n= 732

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 732 296 0 (0.5956284 0.4043716)
    2) sex=male 462  89 0 (0.8073593 0.1926407)
      4) age>=14.25 428  73 0 (0.8294393 0.1705607) *
      5) age< 14.25 34  16 0 (0.5294118 0.4705882)
       10) pclass>=2.5 25   7 0 (0.7200000 0.2800000)
         20) fare>=17.34375 15   0 0 (1.0000000 0.0000000) *
         21) fare< 17.34375 10   3 1 (0.3000000 0.7000000) *
       11) pclass< 2.5 9   0 1 (0.0000000 1.0000000) *
    3) sex=female 270  63 1 (0.2333333 0.7666667)
      6) pclass>=2.5 110  53 1 (0.4818182 0.5181818)
       12) fare>=23.0875 14   2 0 (0.8571429 0.1428571) *
       13) fare< 23.0875 96  41 1 (0.4270833 0.5729167)
         26) age>=16.5 73  36 1 (0.4931507 0.5068493)
           52) fare>=7.72915 63  29 0 (0.5396825 0.4603175)
            104) fare< 15.7 49  19 0 (0.6122449 0.3877551) *
            105) fare>=15.7 14   4 1 (0.2857143 0.7142857) *
           53) fare< 7.72915 10   2 1 (0.2000000 0.8000000) *
         27) age< 16.5 23   5 1 (0.2173913 0.7826087) *
      7) pclass< 2.5 160  10 1 (0.0625000 0.9375000) *
```

```
survived
    0.13 when sex is   male & pclass >= 2 & age >=  9.5
    0.18 when sex is female & pclass >= 3                & fare >=         20.8
    0.23 when sex is   male & pclass >= 2 & age <   9.5 & fare >=         29.1
    0.28 when sex is   male & pclass <  2 & age >= 36.5
    0.32 when sex is female & pclass >= 3 & age >= 17.5 & fare is  7.7 to 10.8
    0.59 when sex is   male & pclass <  2 & age <  36.5
    0.67 when sex is female & pclass >= 3 & age >= 17.5 & fare is 10.8 to 20.8
    0.68 when sex is   male & pclass >= 2 & age <   9.5 & fare <  29.1
    0.75 when sex is female & pclass >= 3 & age >= 17.5 & fare <   7.7
    0.77 when sex is female & pclass >= 3 & age <  17.5 & fare <  20.8
    0.93 when sex is female & pclass <  3
```

<사진 1>- 사전 가지치기 모델의 rule set 추출

과제 7

- 과제 7 – B 추출된 rule 중 가장 중요한 rule 은 무엇인가
  - <사진1>을 통해 sex is male& pclass>=1.5& age>=9.5 에 해당하는 노드를 따라 갈때에 전체의 44%가 결정된다.
  - <사진 2>에서 가장 첫번째 Rule이 추출된 rule에서 가장 중요한 rule로 볼 수 있다.



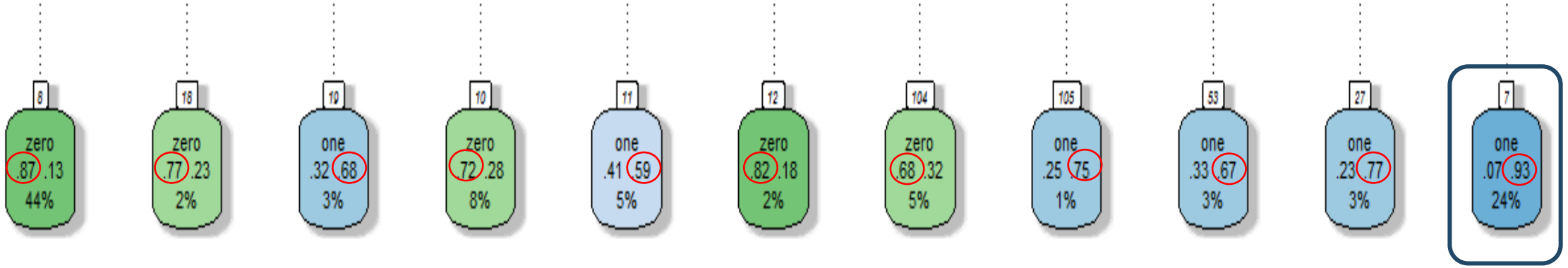
<사진 1>- 의사결정트리

```
survived
    0.13 when sex is   male & pclass >= 2 & age >=  9.5
    0.18 when sex is female & pclass >= 3                & fare >=         20.8
    0.23 when sex is   male & pclass >= 2 & age <   9.5 & fare >=         29.1
    0.28 when sex is   male & pclass <  2 & age >= 36.5
    0.32 when sex is female & pclass >= 3 & age >= 17.5 & fare is  7.7 to 10.8
    0.59 when sex is   male & pclass <  2 & age <  36.5
    0.67 when sex is female & pclass >= 3 & age >= 17.5 & fare is 10.8 to 20.8
    0.68 when sex is   male & pclass >= 2 & age <   9.5 & fare <  29.1
    0.75 when sex is female & pclass >= 3 & age >= 17.5 & fare <   7.7
    0.77 when sex is female & pclass >= 3 & age <  17.5 & fare <  20.8
    0.93 when sex is female & pclass <  3
```

<사진 2>- 추출된 Rule

과제 7

- 과제 7 – C 추출된 rule 중 가장 신뢰도가 높은 rule은 무엇인가
  - <사진1>을 보면 Rule set을 시각적으로 잘 확인할 수 있다. 이때, 중요도를 나타내는 %값 위에 소수점으로 각 Rule의 신뢰도를 측정할 수 있다.
  - 여기서 주목할 점은 one으로 변환시킨 목표변수 값은 오른쪽에 있는 값, zero로 변환시킨 목표변수 값은 왼쪽의 있는 값으로 신뢰도를 판별할 수 있다.
  - 이때, 가장 오른쪽 끝에 있는 Leaf 노드의 정확도는 0.93으로 가장 높게 나타난다. 따라서 해당 Rule이 가장 신뢰도가 높다는 것을 알 수 있다.



<사진 1>- 의사결정트리의 leaf 노드

고찰

- 고찰

  - dataset 분석 과정에서 data의 편중을 발견하고 이로 인해 목표변수 예측에 특이 값 또는 정확한 예측을 방해할 것 이라 생각해 목표변수에 가중치를 두고 training set과 test set을 구성하려 했으나, 이러한 방식은 목표변수를 이진적으로 만들 수 없기에 다른 방법 구상
  이후 dataset에서 data의 비율을 맞추어 삭제하는 방법사용 하려 했으나,
  data set의 data양의 감소를 가져와 정확한 예측이 힘들어진다. 또한 일반적인 dataset이 아니여서 data분석과 이를 통해 얻을 수 있는 일반적 사례와의 연관성이 떨어져 활용도가 낮을거 같다는 생각을 하였다.