Machine Learning

두 가지 데이터셋을 이용한 다양한 머신 러닝 사용

제출일	2021, 04, 26
작성자	정승호

목 차

개요

이 레포트는 사례연구#5로서, 제시된 과제에 대한 사례연구 내용을 포함한다.

사례연구에 사용된 소스파일은 R 파일로 첨부하였으며, 사용된 패키지와 함수명은 파일에 포함되어 있다.

1. 위스콘신 유방암 데이터셋에 분류기법 3개를 적용하여 기법별 결과를 비교.

- 목적변수는(종속변수)는 diagnosis로, Benign(양성), Malignancy(악성)으로 분류한다.

1-1) 의사결정트리 모형

아래와 같이 의사결정트리를 사용해 총 398개의 트리를 생성하였고 이를 통해 분류를 실시하였다.

의사결정트리를 통해 변수들이 암 진단에 미치는 영향력 또한 확인이 가능하다.

```
Call:
rpart(formula = diagnosis ~ ., data = train)
  n= 398

          CP nsplit rel error    xerror       xstd
1 0.77241379      0 1.0000000 1.0000000 0.06621170
2 0.06896552      1 0.2275862 0.3379310 0.04520653
3 0.04827586      2 0.1586207 0.2620690 0.04043277
4 0.01000000      3 0.1103448 0.2137931 0.03687265

Variable importance
   points_worst perimeter_worst     points_mean    radius_worst  concavity_mean concavity_worst      area_worst       area_mean
             18              15              14              14              13              12               3               2
    radius_mean  perimeter_mean    texture_mean   texture_worst      texture_se
              2               2               1               1               1


> rpart.diag$variable.importance #각 변수가 암진단에 미치는 영향력 확인가능
   points_worst perimeter_worst     points_mean    radius_worst  concavity_mean concavity_worst      area_worst       area_mean
     124.437729      102.364416      100.978157      100.620447       94.858269       85.678436       18.500232       16.958546
    radius_mean  perimeter_mean    texture_mean   texture_worst      texture_se    perimeter_se       radius_se  symmetry_mean
      16.187703       14.646017        9.182186        7.870445        5.246964        2.623482        2.623482        2.623482
```

rpart.pred2	1	2
B	100	3
M	4	64

또한, 분류기법의 분류 정확도 확인을 위해 혼돈 매트릭스 모형을 생성하였다. 이는 마지막 그림에서 확인할 수 있으며, 현재 예측값은 데이터 샘플링과정에서 검정데이터의 B가 1, M이 2로 변환되어 있다. 이 혼돈 매트릭스 모형을 통해 의사결정트리 모형의 분류 정확도를 계산하면 약 95%의 분류 정확도가 산출된다.

1-2) 인공 신경망 모형

아래와 같이 인공 신경망 모형 생성에 앞서 전처리로서 2차로 데이터 샘플링을 실시했으며 이후 인공 신경망 모형을 통해 분류를 실시하였다.

```
# 2차 데이터 샘플링
# 인공신경망에 사용할 수 있도록 diagnosis의 인자값을 문자에서 숫자로 변환
train$diagnosis[train$diagnosis == 'B'] <- 1
train$diagnosis[train$diagnosis == 'M'] <- 2
test$diagnosis[test$diagnosis == 'B'] <- 1
test$diagnosis[test$diagnosis == 'M'] <- 2
train$diagnosis <- as.numeric(train$diagnosis) #캐릭터값에서 변경
test$diagnosis <- as.numeric(test$diagnosis)

# 정규화 함수
normalize <- function(x){
  return((x-min(x))/(max(x)-min(x)))}
train.norm <- as.data.frame(sapply(train, normalize))
test.norm <- as.data.frame(sapply(test, normalize))
```

```
nnet.diag1 <- nnet(diagnosis ~ ., data = train.norm, size = 1) #33개 가중치 확인
nnet.diag2 <- nnet(diagnosis ~ ., data = train.norm, size = 2) #65개 가중치
nnet.diag3 <- nnet(diagnosis ~ ., data = train.norm, size = 3) #97개 가중치

summary(nnet.diag1) #가중치 수치 확인 가능
summary(nnet.diag2)
summary(nnet.diag3)

# 가중치가 너무 많을경우 연산량이 증가하므로 여기서는 은닉노드층이 2인 쪽을 사용해 작업 수행
```

```
> summary(nnet.diag2)
a 30-2-1 network with 65 weights
options were -
  b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9->h1 i10->h1 i11->h1 i12->h1 i13->h1 i14->h1 i15->h1
  86.71    6.81   27.84    7.42   -3.08   24.19   29.82  -61.22  -61.47   -2.02    6.71  -51.37   41.99  -11.60  -43.54  -29.09
i16->h1 i17->h1 i18->h1 i19->h1 i20->h1 i21->h1 i22->h1 i23->h1 i24->h1 i25->h1 i26->h1 i27->h1 i28->h1 i29->h1 i30->h1
  76.57   10.07   -5.39   15.34   44.37  -23.56  -93.50   -9.65  -33.52  -25.92    7.86  -59.41  -29.06  -42.32   -0.81
  b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9->h2 i10->h2 i11->h2 i12->h2 i13->h2 i14->h2 i15->h2
  -1.66   -1.76   -0.54   -0.96   -1.18   -0.67   -0.63   -1.06   -1.05   -0.89   -0.03   -1.64    0.19   -1.80   -1.32   -0.27
i16->h2 i17->h2 i18->h2 i19->h2 i20->h2 i21->h2 i22->h2 i23->h2 i24->h2 i25->h2 i26->h2 i27->h2 i28->h2 i29->h2 i30->h2
  -0.03    0.44   -1.25   -0.04   -0.80   -1.53   -1.61   -1.42   -0.96   -1.21   -1.55   -1.79   -2.22   -0.05   -0.51
   b->o   h1->o   h2->o
  40.31 -100.24    0.25
```

은닉 노드를 1부터 3까지 바꿔서 분류를 실시하였으며, 연산량이 갈수록 증가했기 때문에 적당한 값으로

2개의 은닉 노드를 가지는 인공 신경망으로 분류를 실시하였으며 분류과정에서 산출된 가중치들은 위와 같다.

```
nnet.pred   0   1
        0 103   6
        1   1  61
```

또한 이 기법 역시 분류 정확도 확인을 위해 혼돈 매트릭스 모형을 생성하였다. 이 혼돈 매트릭스 모형을 통해 인공 신경망 분석기법의 분류 정확도를 계산하면 앞의 기법과 흡사하게 약 95%의 분류 정확도가 산출된다.

1-3) 랜덤 포레스트 모형

마지막 분류기법으로 랜덤 포레스트 모형을 적용하였으며, 분류를 위해 데이터 전처리가 한번 더 필요했기 때문에 이 또한 실시하였다.

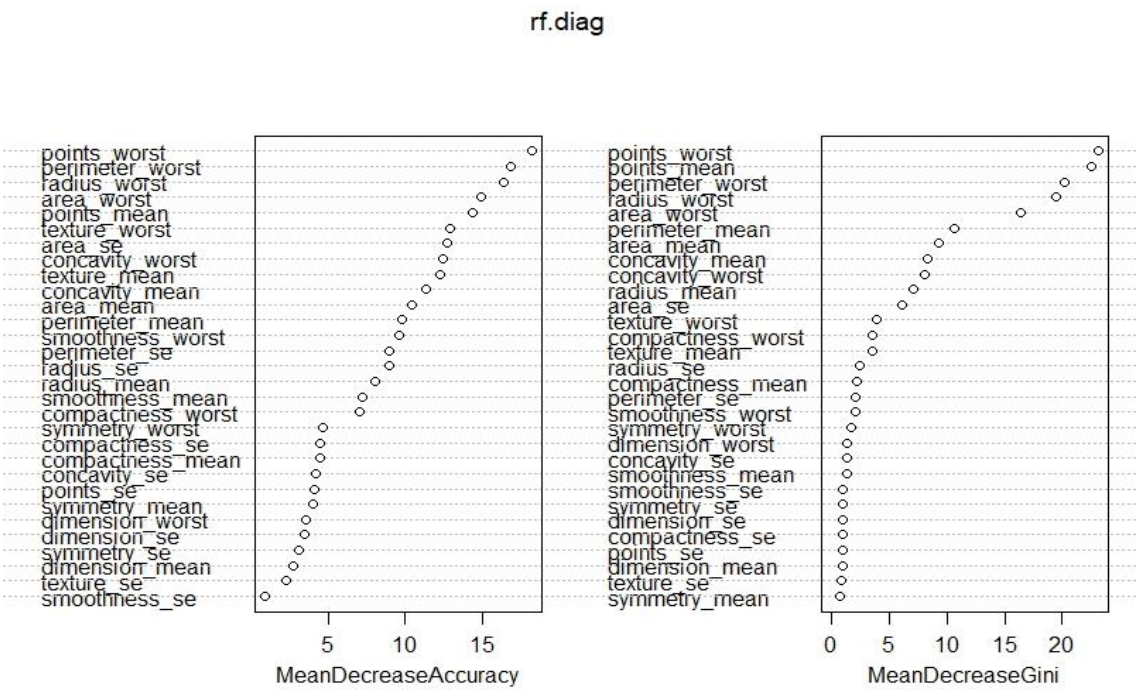
```
#분류용 랜덤포레스트 사용을 위해 목적변수인 diagnosis를 요인(factor)화 할것
train$diagnosis <- as.factor(train$diagnosis)
test$diagnosis <- as.factor(test$diagnosis)
# 분류 목적일 때는 m=sqrt(p) 개의 설명변수를 사용하는것이 일반적임.
str(train) #31개 변수중 목적변수를 제외하면 30.
sqrt(30) #5.4개이므로 변수는 5개로 사용
```

위와 같이 요인화를 거쳐 5개의 설명변수로 모형을 생성해 분류분석을 실행했으며 결과는 다음과 같다.

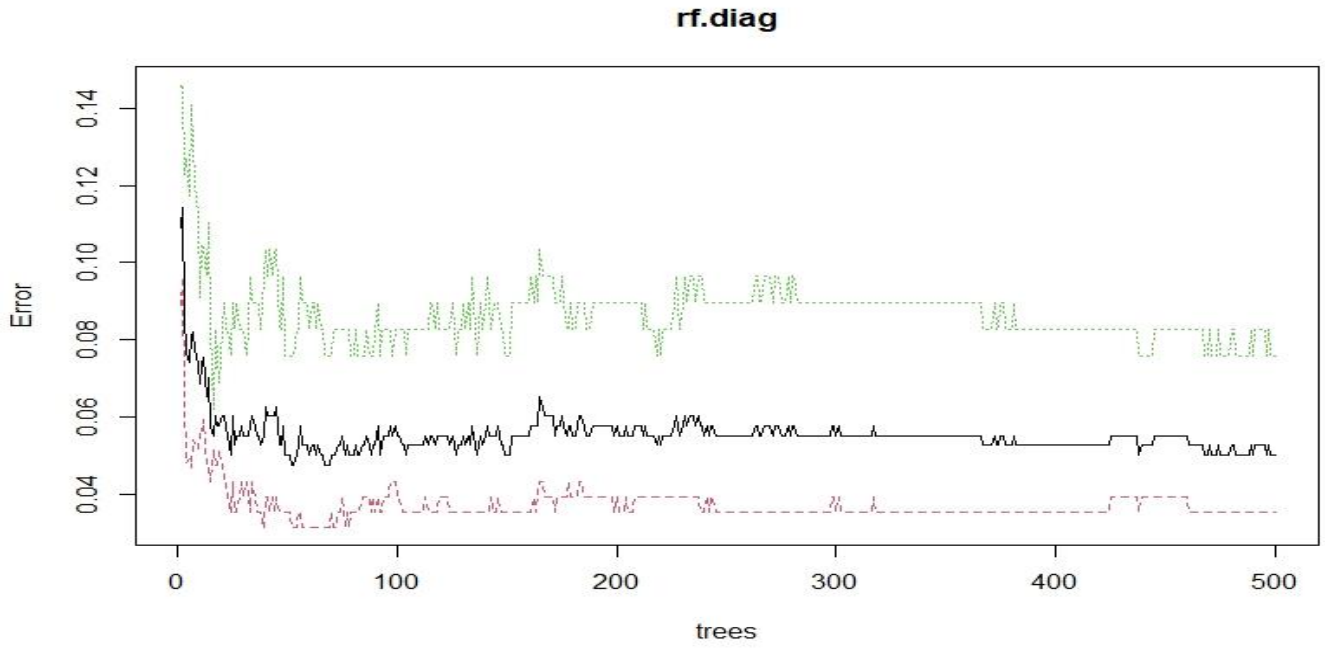
```
Call:
 randomForest(formula = diagnosis ~ ., data = train, mtry = 5,      importance = T)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 5

        OOB estimate of  error rate: 5.03%
```

	1	2	MeanDecreaseAccuracy	MeanDecreaseGini
points_worst	14.2402678	11.2929808	18.2002259	23.1276612
points_mean	9.4703820	11.8610648	14.3990289	22.5421210
perimeter_worst	13.6496556	11.0683915	16.8089994	20.1941322
radius_worst	14.0912131	10.3137039	16.3651013	19.5263612
area_worst	12.2492161	10.4362701	14.9180474	16.3700864
perimeter_mean	8.0049201	6.0232345	9.8315548	10.6393038

500개의 트리와 5개의 변수로 분석을 실행하였으며, OOB는 5.03%로 나타났다. 데이터 내에서 중요한 변수 또한 확인할 수 있으며, 아래 시각화 자료를 통해 특히 영향을 크게 미치는 2개 변수를 확인할 수 있다.



트리 숫자에 따른 오차율 또한 시각화 자료로 첨부하며, 이를 통해 약 250개 이상의 트리를 생성해 분류분석을 실시하였을 때 오차율이 확연히 안정화되는 모습을 확인할 수 있으며, 분석에 사용된 500개의 트리의 경우 그보다 더욱 안정화된 상태임을 알 수 있다.




```
rf.pred   B   M
      1 103   1
      2   1  66
```

마지막으로, 랜덤 포레스트 모형 역시 예측치와 검정데이터를 이용해 혼돈 매트릭스 모형을 생성하였다. 랜덤 포레스트 모형의 분류 정확도를 계산하면 약 98.5% 이상의 정확도가 산출되며, 이를 통해 세 가지 분류기법중 랜덤 포레스트 모형을 사용해 분류 분석을 실시했을 때 해당 데이터셋의 분류 정확도가 가장 뛰어나다는 것을 알 수 있다. 때문에, 위에서 수행한 세 가지 분류분석 기법 중에는 랜덤 포레스트 모형을 사용하는 것이 바람직하다.

(2) mlbench 패키지 내 BostonHousing 데이터셋을 대상으로 예측기법 3개를 적용하여 기법별 결과를 비교

 # 종속변수는MEDV 또는CMEDV를사용

2-1) 다중회귀분석 모형

주어진 데이터셋을 대상으로 다중회귀분석을 통해 수치 예측을 시행하였으며 분석결과는 다음과 같다.

```
Call:
lm(formula = medv ~ ., data = train.Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-13.1849  -2.9870  -0.7744   1.6318  24.5109

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  47.814134   6.638115   7.203 3.82e-12 ***
crim         -0.112097   0.037975  -2.952 0.003378 **
zn            0.052563   0.017849   2.945 0.003453 **
indus         0.006241   0.078040   0.080 0.936310
chas1         2.121665   1.122889   1.889 0.059679 .
nox         -19.716044   5.044212  -3.909 0.000112 ***
rm            2.931967   0.514002   5.704 2.54e-08 ***
age           0.003114   0.017219   0.181 0.856589
dis          -1.723664   0.271859  -6.340 7.30e-10 ***
rad           0.325645   0.081700   3.986 8.23e-05 ***
tax          -0.012649   0.004605  -2.747 0.006337 **
ptratio      -1.074939   0.176122  -6.103 2.83e-09 ***
b             0.006221   0.003634   1.712 0.087887 .
lstat        -0.570708   0.065029  -8.776  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.108 on 340 degrees of freedom
Multiple R-squared:  0.7142,    Adjusted R-squared:  0.7033
F-statistic: 65.37 on 13 and 340 DF,  p-value: < 2.2e-16
```

주어진 변수들로 다중회귀분석을 실시한 바, 설명력은 0.71로 나타나며 P값 역시 유의수준모다 낮게 나타나기 때문에 모형 자체는 유효하지만, F 검정통계량이 65로 조금 낮게 나타나고, 몇몇 회귀계수들이 유의미한 결과를 보여주지 못하고 있기 때문에, 전진후진법을 통해 분석모델을 좀 더 교정하여 다시금 분석을 실시한다.

```
Call:
lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
    tax + ptratio + b + lstat, data = train.Boston)

Residuals:
     Min       1Q   Median       3Q      Max
-13.2389  -3.0047  -0.7988   1.6404  24.5719

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  47.709290   6.591677   7.238 3.03e-12 ***
crim         -0.112475   0.037802  -2.975  0.00313 **
zn            0.052086   0.017592   2.961  0.00328 **
chas1         2.147600   1.109289   1.936  0.05369 .
nox         -19.377584   4.702626  -4.121 4.75e-05 ***
rm            2.946181   0.500806   5.883 9.61e-09 ***
dis          -1.742795   0.253580  -6.873 2.99e-11 ***
rad           0.323027   0.077915   4.146 4.27e-05 ***
tax          -0.012460   0.004045  -3.081  0.00223 **
ptratio      -1.070739   0.173761  -6.162 2.02e-09 ***
b             0.006230   0.003623   1.719  0.08645 .
lstat        -0.566436   0.061173  -9.260  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.093 on 342 degrees of freedom
Multiple R-squared:  0.7142,    Adjusted R-squared:  0.705
F-statistic:  77.7 on 11 and 342 DF,  p-value: < 2.2e-16
```

전진후진법을 통한 교정결과 설명력 자체는 수정값이 약간 상향된 정도지만, F 검정통계량이 77로 상승하였으며 회귀계수들의 t값과 유의수준도 훨씬 나아진 형태로 모델이 구성되었다. 이제 이 모델을 통해 수치 예측을 실시하고, 실제 값과의 평균 제곱 오차(MSE)를 계산하여 모델의 수치 예측력을 확인할 것이다.

```
lm.pred <- predict(lm.Boston.fit, newdata = test.Boston)
lm.pred <- sqrt(mean((lm.pred - test.Boston$medv)^2))
```

다음과 같이 계산한 결과, 다중회귀분석 모델의 MSE는 약 3.966으로 예측 수치와 실제값이 약 4정도의 차이가 발생하는 것으로 확인되었다.

2-2) SVM(서포트 벡터 머신)

다음으로 서포트 벡터 머신을 사용하여 수치 예측을 시행하고, MSE를 산출해 실제값과의 오체를 확인할 것이다.

```
svm.Boston.G <- ksvm(medv ~ ., data = train.Boston) #기본 가우시안 함수 사용.
svm.Boston.G
svm.Boston.L <- ksvm(medv ~ ., data = train.Boston, kernal = "vanila dot") #리니어 함수 사용
svm.Boston.L
svm.Boston.P <- ksvm(medv ~ ., data = train.Boston, kernal = "poly dot") #polynomial 함수 사용
svm.Boston.P
```

수치 예측을 목적으로 서포트 벡터 머신 모델을 구성해 가동했기 때문에, 위 사진과 같이 커널 트릭을 사용하였으며, 3가지 함수를 사용한 결과를 비교해 가장 많은 서포트 벡터를 구성한 방식을 수치 예측에 사용하였다.

```
Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.0943964383583693

Number of Support Vectors : 243

Objective Function Value : -59.8121
Training error : 0.107241

Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.114788579991383

Number of Support Vectors : 237

Objective Function Value : -58.7632
Training error : 0.099779

Support Vector Machine object of class "ksvm"

SV type: eps-svr  (regression)
 parameter : epsilon = 0.1  cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.100594462922717

Number of Support Vectors : 238

Objective Function Value : -59.4339
Training error : 0.104664
```

결과는 다음과 같으며 가우시안 함수를 사용한 값이 이번 분석결과에서는 243개의 서포트 벡터를 구성했디 때문에 해당 모델을 수치예측에 사용하였다.

```
svm.pred <- predict(svm.Boston.G, newdata = test.Boston)
svm.MSE <- sqrt(mean((svm.pred - test.Boston$medv)^2))
```

위와 같이 예측치를 먼저 구성한 다음 2-1번에서와 같이 예측값과 실제값의 MSE를 계산하였으며, 그 결과 3.036으로 실제값과는 약 3정도의 차이가 발생하는것으로 확인되었다.

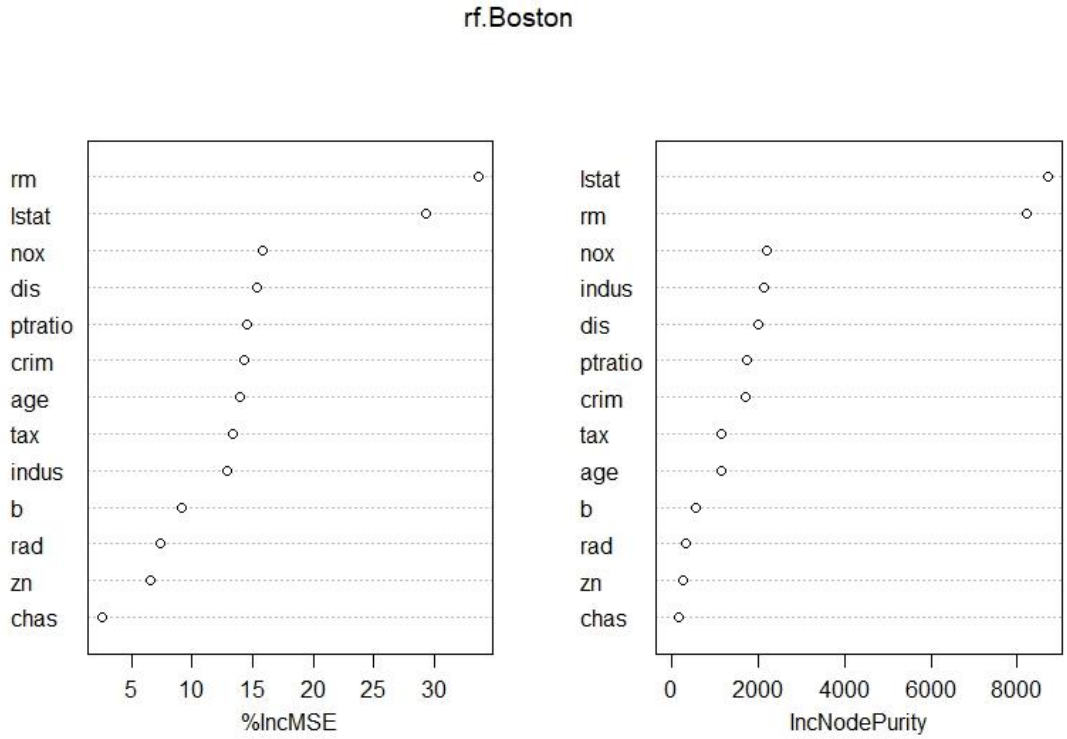
2-3) 랜덤 포레스트 모형

마지막으로 1-3에서 사용했던 랜덤 포레스트 모형을 이번에는 수치 예측을 위해 사용하였으며, 수치 예측의 용도로 사용되었기 때문에 1-3과 같은 전처리 과정 없이 분석을 실시하였다. 설명변수는 수치 예측 목적의 경우 일반적으로 총 설명변수/3을 사용하므로, 데이터셋이 제공하는 14개의 변수 중 목적변수 1개를 제외하고 13/3 = 4.3333..이므로 4개의 변수를 지정하여 분석을 실시하였다.

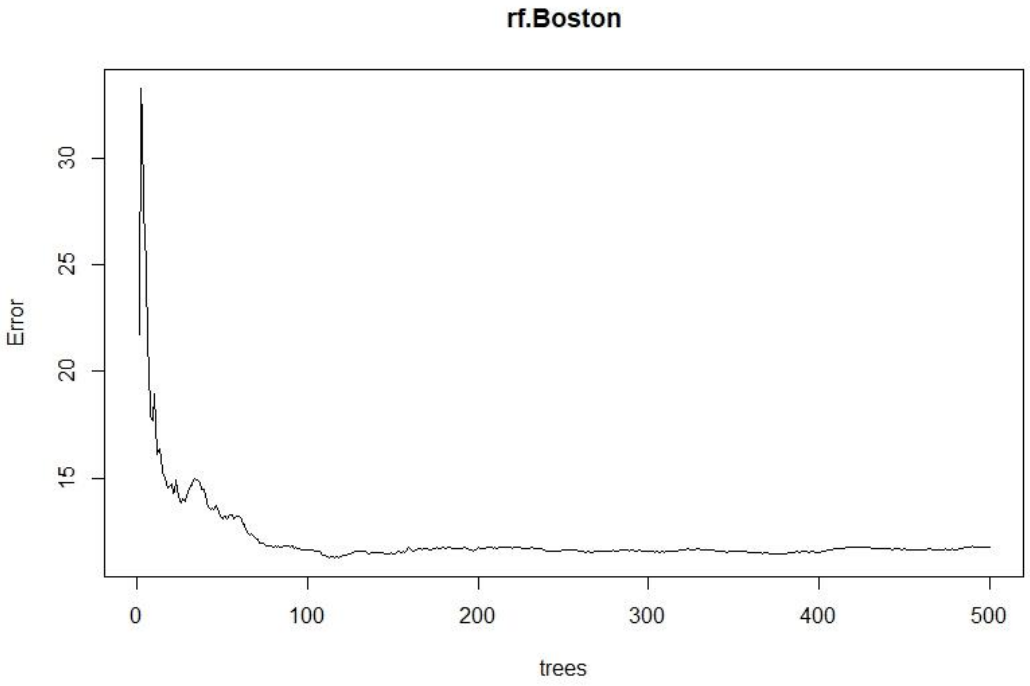
```
Call:
 randomForest(formula = medv ~ ., data = train.Boston, mtry = 4,      importance = T)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 11.77546
                    % Var explained: 86.57
```

```
          %IncMSE IncNodePurity
crim    14.353524     1713.4204
zn       6.512220      251.7735
indus   12.938558     2124.5623
chas     2.591163      161.7133
nox     15.811182     2203.2692
rm      33.594368     8207.9895
age     13.907043     1139.7265
dis     15.404457     2018.5541
rad      7.419896      319.5016
tax     13.389853     1140.5447
ptratio 14.500578     1753.5583
b        9.141859      563.2517
lstat   29.270210     8701.6763
```

분석을 시행한 결과 약 11.77의 잔차를 가진 랜덤 포레스트 모델이 생성되었으며, 각 변수가 가지는 중요도 또한 위와 같이 확인할 수 있다. 이 중 가장 중요도가 높은 변수와 트리의 안정도에 관한 시각화는 다음 장에 첨부되었다.



위와 같이 목적변수에 가장 큰 영향을 미치는 설명변수인 rm과 lstat를 확인할 수 있다.



트리 숫자가 약 50개를 넘어가면서 오차가 확연하게 안정되는 모습을 확인할 수 있다.

```
rf.pred <- predict(rf.Boston, newdata = test.Boston)
rf.MSE <- sqrt(mean((rf.pred - test.Boston$medv)^2))
```

랜덤 포레스트 모델의 분석 결과를 확인했으므로, 최종적으로 예측치를 생성하고 MSE를 계산하였다.
그 결과 약 2.848이 산출되었으며, 이는 세 가지 수치 예측 방법 중 가장 실제값과 오차율이 적다. 때문에, 데이터셋의 값을 예측할 때는 랜덤 포레스트 모델을 사용하는 것이 바람직하게 보인다.