사례연구 4

머신러닝기반
데이터 분석 보고서

B3조 유준 문현진 임성현

2021년 11월 21일

□ 목차

1. 서론

2. 본론

1) 위스콘신 유방암 데이터 분류 분석

(1) 데이터 셋 확인
(2) 의사 결정 나무 기법
(3) Naive Bayes 기법
(4) Random Forest 기법
(5) 각 분류 분석의 비교

2) Abalone 데이터 예측 분석

(1) 데이터 셋 확인
(2) Random Forest기법
(3) 다중 회귀 분석
(4) 각 예측 분석의 비교

3) K-means Clustering 시각화

(1) 오차 제곱합의 그래프
(2) K-means Clustering 시각화

3. 결론

4. 부록

5. 참고자료

1. 서론

이 보고서의 목적은 학습한 분류기법과 예측기법, 군집분석 등을 각 연구주제에 활용하여 실습을 진행하고 그에 따른 결과를 알아보는 것에 있다.

○ 위스콘신 유방암 데이터 셋을 대상으로 2개의 분류 기법을 적용하여 각 기법 결과를 비교한다.

○ Abalone DataSet을 대상으로 전복의 나이를 예측하고, 2가지의 예측기법을 사용하여 결과를 비교한다.

○ 내장 DataSet ″ iris″ 를 대상으로 K-means Clustering을 실행하고 시각화한다.

2. 본론

1) 위스콘신 유방암 데이터 분류 분석

(1) DataSet 확인

Id	Diagnosis	Radius_mean	Texture_mean	…..	Dimesion_worst
87139402	B	12.32	12.39	……	0.06771
8910251	B	10.6	18.95	……	0.07587
905520	B	11.04	16.83	……	0.07881
868871	B	11.28	13.39	……	0.06784
9012568	B	15.19	13.21	……	0.06766
906539	B	11.57	19.04	……	0.08284
925291	B	11.51	23.93	……	0.08732
87880	M	13.81	23.75	……	0.1086
862989	B	10.49	19.29	……	0.07552
89827	B	11.06	14.96	……	0.0908
91485	M	20.59	21.24	……	0.08999
8711003	B	12.25	17.94	……	0.08132
9113455	B	13.14	20.74	……	0.08174
857810	B	13.05	19.31	……	0.06289
9111805	M	19.59	25	……	0.06091
925277	B	14.59	22.68	……	0.08004
867387	B	15.71	13.93	……	0.07071
89511502	B	12.67	17.3	……	0.06888

↳ 종양의 크기, 모양 등 다양한 속성 값을 기반으로 해당 종양이 악성인지 양성인지 분류하는 DataSet이다.

(2) 의사결정 나무기법

rpart함수를 이용하여 의사결정 나무 모형을 훈련하고, predict함수를 이용하여 유방암 데이터를 Diagnosis 기준으로 분석한 결과이다.

의사결정 나무 분류분석 결과표

	실측 값	
예측 값	B(양성)	M(악성)
B(양성)	**101**	8
M(악성)	7	**55**

↳ 분류 결과에서 실제로는 양성이지만 악성으로 7개가 잘못 분류되었고, 악성인데 양성으로 8개가 잘못 분류 되었다.

분류 정확도는 (101+ 55) / (101+ 8+ 7+ 55) 로 0.9123이다.
즉 91%의 정확도로 분류하였다.

(3) Naive Bayes 기법

NaiveBayes 함수를 이용하여 나이브 베이즈 모형을 훈련하고, predict 함수를 이용하여 유방암 데이터를 Diagnosis 기준으로 분석한 결과이다.

Naive Bayes 분류분석 결과표

	실측 값	
예측 값	B(양성)	M(악성)
B(양성)	**102**	3
M(악성)	6	**60**

↳ 분류 결과에서 실제로는 양성이지만 악성으로 6개가 잘못 분류되었고, 악성인데 양성으로 3개가 잘못 분류되었다.

분류 정확도는 (102+ 60) / (102+ 3+ 6+ 60) 로 0.9474이다.
즉 94%의 정확도로 분류하였다.

(4) Random Forest 기법

RandomForest 함수를 이용하여 Random Forest 모형을 훈련하고, predict 함수를 이용하여 유방암 데이터를 Diagnosis 기준으로 분석한 결과이다.

Random Forest 분류분석 결과표

	실측 값	
예측 값	B(양성)	M(악성)
B(양성)	**106**	2
M(악성)	2	**61**

↳ 분류 결과에서 실제로는 양성이지만 악성으로 2개가 잘못 분류되었고, 악성인데 양성으로 2개가 잘못 분류되었다.

분류 정확도는 (106+ 61) / (106+ 2+ 2+ 61) 로 0.9766이다.
즉 97%의 정확도로 분류하였다.

(5) 각 분류 분석의 비교

의사결정 나무 기법의 정확도는 91%
Naïve Bowes 기법의 정확도는 94%
Random Forest 기법의 정확도는 97% 로

3가지의 분류 기법에서 Random Forest 기법의 성능이 제일 우수하다.

2) Abalone 데이터 예측분석

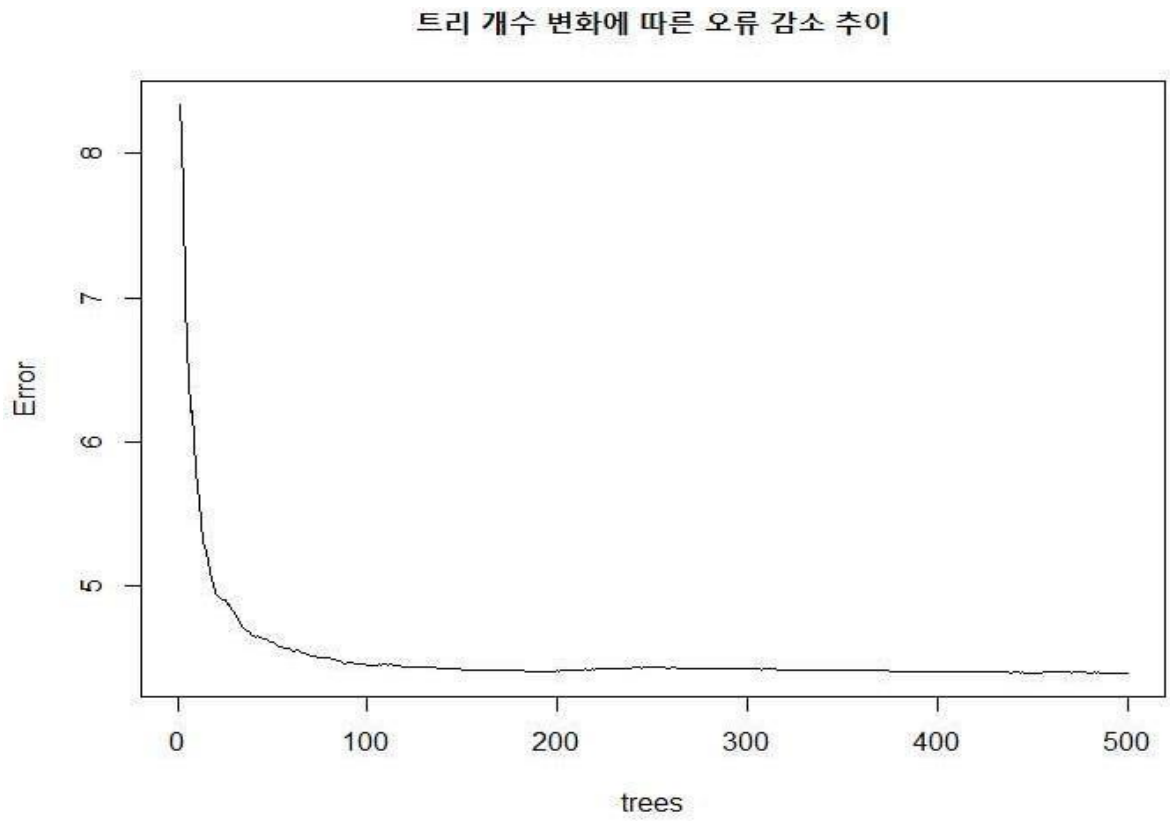
(1) DataSet 확인

Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15
M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7
F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9
M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10
I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7
I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8
F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20
F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16
M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9
F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19
F	0.525	0.38	0.14	0.6065	0.194	0.1475	0.21	14
M	0.43	0.35	0.11	0.406	0.1675	0.081	0.135	10
M	0.49	0.38	0.135	0.5415	0.2175	0.095	0.19	11
F	0.535	0.405	0.145	0.6845	0.2725	0.171	0.205	10
F	0.47	0.355	0.1	0.4755	0.1675	0.0805	0.185	10
M	0.5	0.4	0.13	0.6645	0.258	0.133	0.24	12
I	0.355	0.28	0.085	0.2905	0.095	0.0395	0.115	7

↳ 전복의 성별, 길이, 지름 등 다양한 속성값을 기반으로 해당 전복의 나이를 측정한 DataSet이다.

(2) Random Forest 기법(1/3)

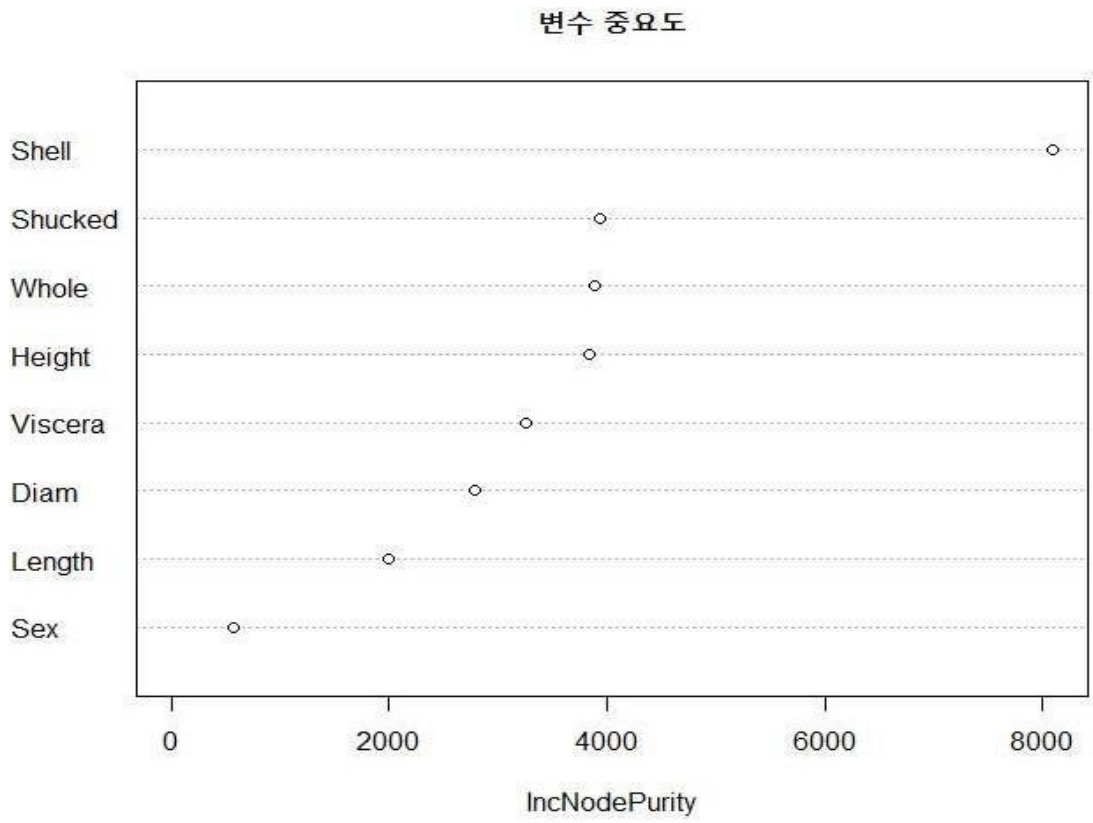
Random Forest기법을 이용하여 Abalone 데이터를 Rings기준으로 분석하고 예측하려고 한다.



↳ RandomForest함수를 이용하여 표시한 그래프로 트리 개수 변화에 따른 오류 감소 추이를 나타낸다.
트리가 100개를 넘어간 뒤로는 비교적 안정된 상태를 유지하는 것을 확인할 수 있다.

(2) Random Forest 기법(2/3)

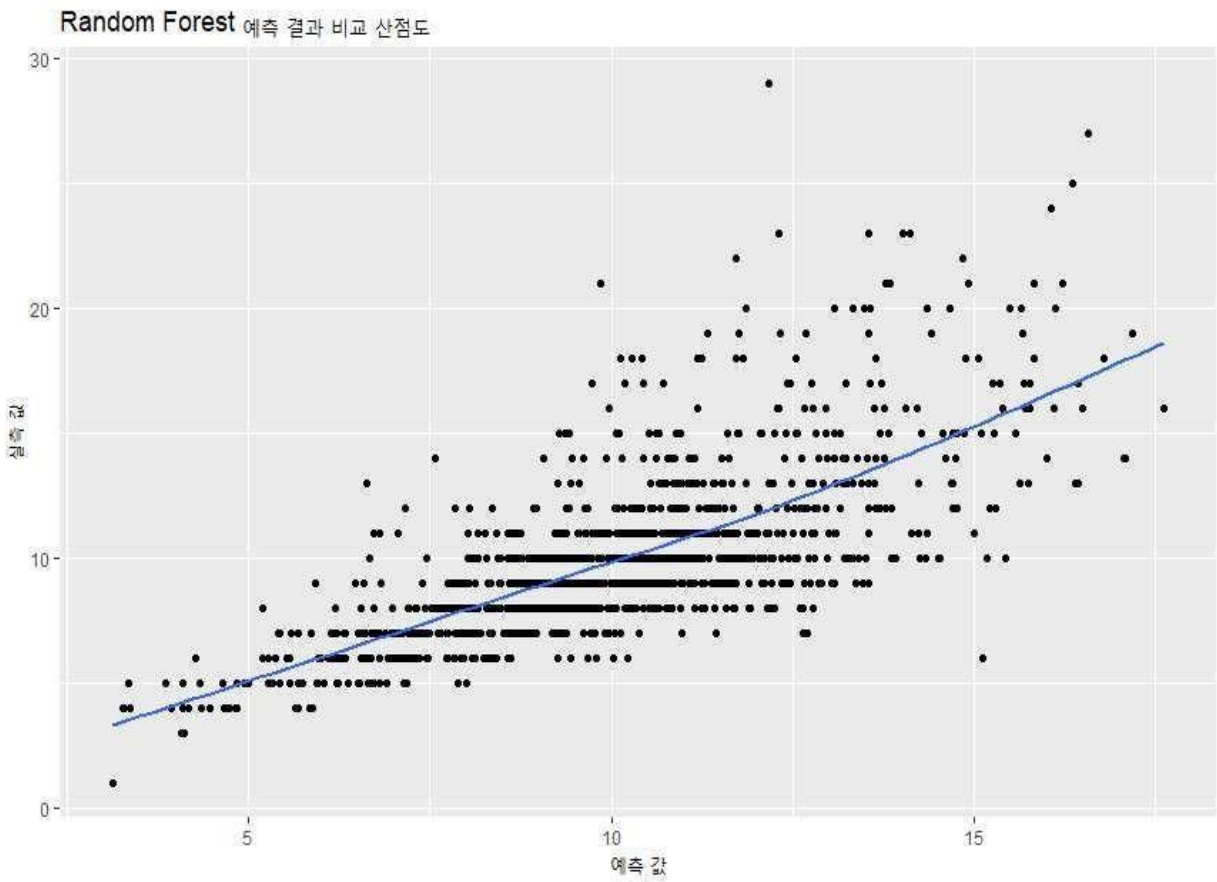
또한 랜덤 포레스트 모형의 특장점 중 하나는 변수의 중요도를 확인할 수 있다는 것이다



↳ 변수 중요도 그래프를 통해서 Rings 변수에 영향을 미치는 설명변수로 Shell 변수가 다른 변수들에 비하여 압도적으로 높은 중요도 측도를 보여주고 있다.

(2) Random Forest 기법(3/3)

Random Forest 모형이 적합되었으므로 이제 평가용 데이터 세트를 이용하여 결과를 예측해보고 예측값과 실제값 간의 평균제곱오차(MSE)와 상관계수를 계산해본다.



↳ 위 그래프는 Random Forest 기법을 사용해 얻은 예측값과 실제값 비교를 나타낸 산점도 그래프다. 두 값은 양의 상관관계를 갖고 있고, 상관계수는 0.7256으로 높은 상관관계를 갖고 있다.
즉 약 73%의 예측율을 보이고 있다고 판단한다.

또한 평가 데이터의 평균 제곱오차는 약 5.22이다.

(3) 다중 회귀분석(1/3)

다중 회귀분석을 이용하여 Abalone 데이터를 Rings 기준으로 나머지 변수를 설명변수로 하여 분석하고 예측하려고 한다.

```
Call:
lm(formula = Rings ~ ., data = train_ab)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5180 -1.3004 -0.2873  0.8668 11.5562

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.49287    0.34084  10.248  < 2e-16 ***
SexI         -0.82326    0.11935  -6.898 6.45e-12 ***
SexM          0.04733    0.09805   0.483    0.629
Length       -0.80968    2.11541  -0.383    0.702
Diam         13.79298    2.58709   5.331 1.05e-07 ***
Height        9.74501    1.66858   5.840 5.79e-09 ***
Whole         8.04423    0.84550   9.514  < 2e-16 ***
Shucked     -19.14192    0.96500 -19.836  < 2e-16 ***
Viscera      -9.79829    1.49349  -6.561 6.32e-11 ***
Shell         8.97856    1.31854   6.809 1.18e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.154 on 2913 degrees of freedom
Multiple R-squared:  0.5436,    Adjusted R-squared:  0.5422
F-statistic: 385.5 on 9 and 2913 DF,  p-value: < 2.2e-16
```

↳ 다중 회귀분석 결과 조정 R-squared값이 0.5422이다.
그러나 회귀계수 중 SexM과 Length는 유의하지 않는 수준의 변수 조정을 통해 다중 회귀분석을 실시한다.

(3) 다중 회귀분석(2/3)

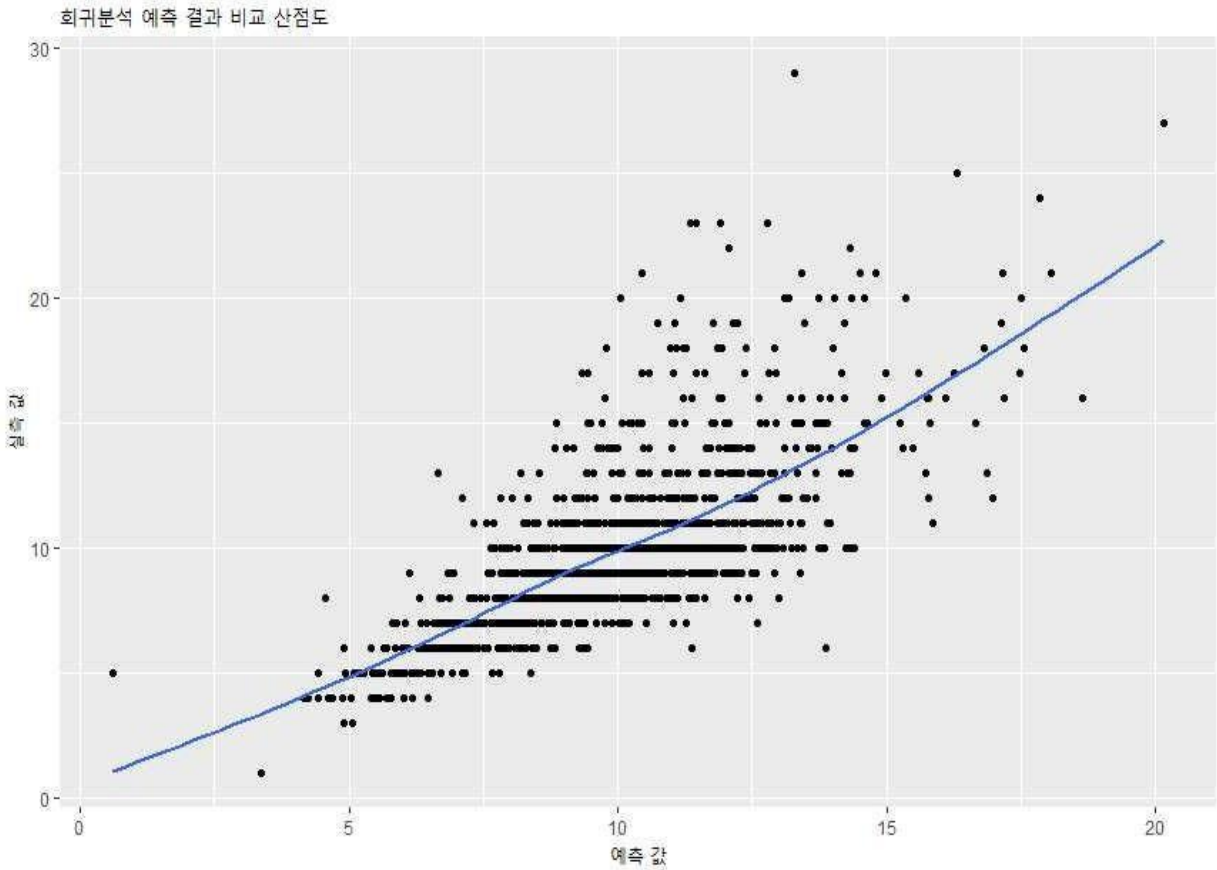
변수 조정 후 다중 회귀분석 재실시 결과는 다음과 같다.

```
Call:
lm(formula = Rings ~ Sex + Diam + Height + Shucked + Shell, data =
 train_ab)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5601 -1.3461 -0.3255  0.9014 11.8072

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.40209    0.32438  10.488  < 2e-16 ***
SexI         -0.85842    0.11972  -7.170 9.47e-13 ***
SexM          0.04070    0.09931   0.410    0.682
Diam         12.90210    1.15109  11.209  < 2e-16 ***
Height        9.74247    1.68961   5.766 8.96e-09 ***
Shucked     -11.81928    0.43648 -27.079  < 2e-16 ***
Shell        18.67266    0.76533  24.398  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.186 on 2916 degrees of freedom
Multiple R-squared:  0.5292,    Adjusted R-squared:  0.5282
F-statistic: 546.2 on 6 and 2916 DF,  p-value: < 2.2e-16
```

↳ 모든 회귀계수들이 유의하게 나타났다. 회귀 적합결과를 이용하여 평가데이터 수치 예측을 시행한다.
predict 함수를 사용하여 회귀 적합의 예측력을 확인한다.

(3) 다중 회귀분석(3/3)

다중 회귀분석을 이용하여 얻은 예측값과 실제값의 비교 산점도 그래프는 다음과 같다.



↳ 예측값과 실제값은 양의 상관관계를 갖고 있으며,
상관계수는 0.7097로 높은 상관관계를 갖고 있다.
따라서 약 71%의 예측률을 보이고 있다고 판단한다.

또한 다중 회귀분석을 이용한 평가 데이터 평균 제곱오차는 약 5.48이다

(4) 각 예측 분석의 비교

각 기법의 성능을 비교하기 위해서 평균제곱오차(MSE)의 값을 구했다.

MSE의 값이 더 낮을수록 기법의 성능이 좋다고 평가한다.

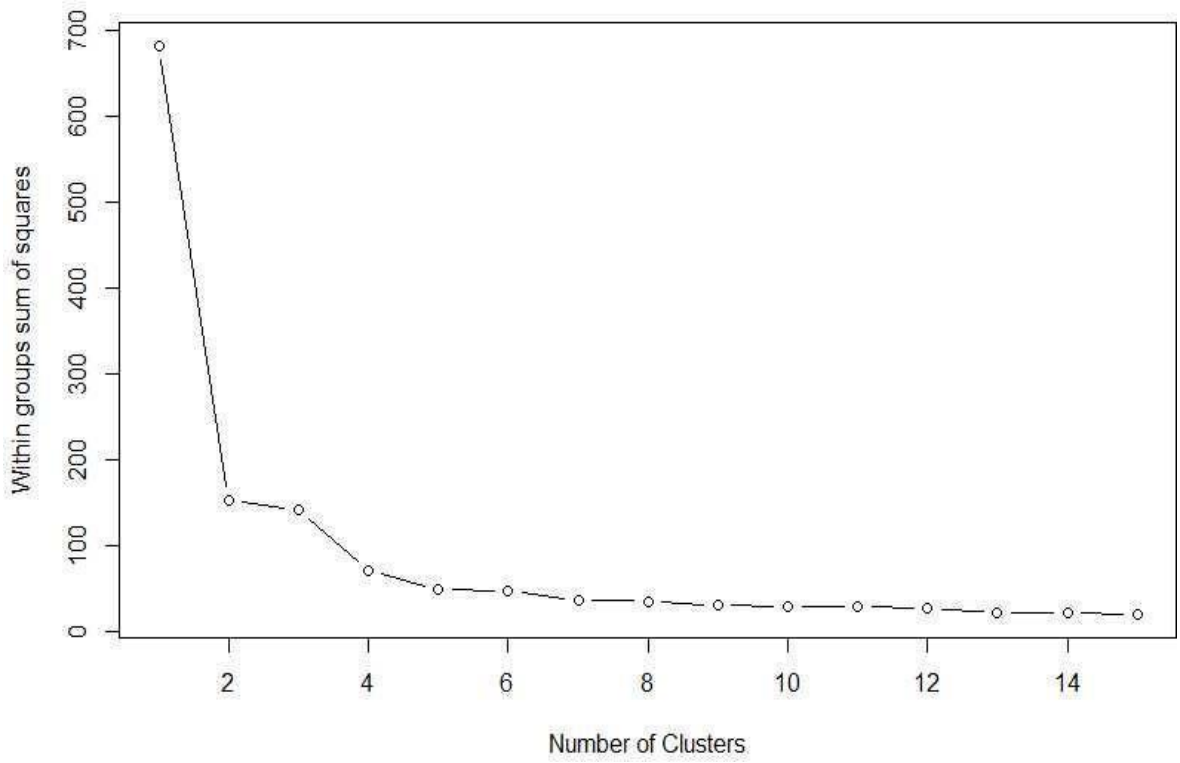
Random Forest MSE : 5.22
다중 회귀분석 MSE : 5.48

따라서 Random Forest 기법의 성능이 더 좋다고 평가한다.

3) K-means Clustering 시각화

(1) 오차 제곱합의 그래프(1/2)

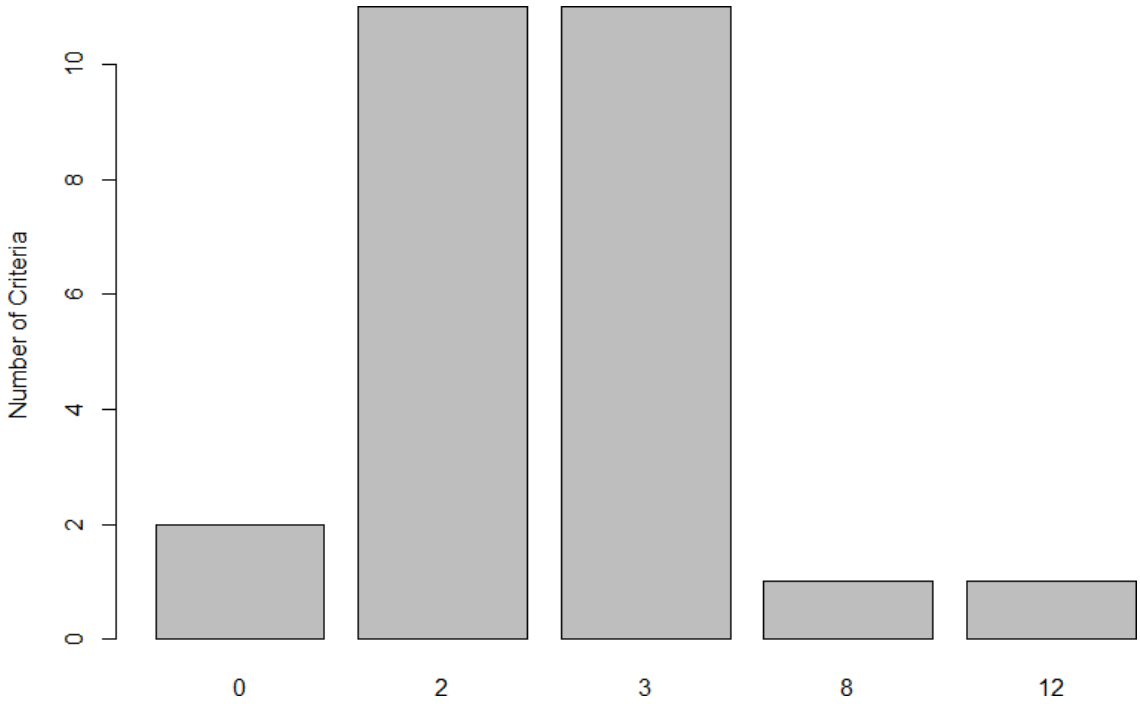
K-means Clustering은 군집의 수를 미리 설정해야 한다.
오차 제곱합의 그래프를 통해서 적절한 군집의 수에 대한 정보를 얻을 수 있다.



↳ 군집수 2에서 오차제곱합이 크게 감소되었다.
즉, 군집수 2개가 적당하다고 판단한다.

(1) 오차 제곱합의 그래프(2/2)

군집수를 파악하는데 사용하는 Nbclust 함수를 사용하여 나타낸 그래프는 다음과 같다.



↳ 최적의 군집수를 정하기 위해 사용되는 지수 가운데 11개의 지수가 2, 3을 최적의 군집수로 투표한 결과를 보여준다.

따라서 두 방법 모두 신뢰하여 군집 수는 2로 설정한다.

(2) K-means Clustering 시각화(1/2)

앞의 수행단계에서 적정한 군집 개수 K = 2으로 설정하였으므로 kmeans 함수를 사용하여 군집화를 수행하고 결과를 확인한다.

↳ 군집화 결과 각 군집들의 중심값($centers)와 각 클러스터의

```
> kmeans_result$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.301031    2.886598     4.958763    1.695876
2     5.005660    3.369811     1.560377    0.290566
> kmeans_result$size
[1] 97 53
```

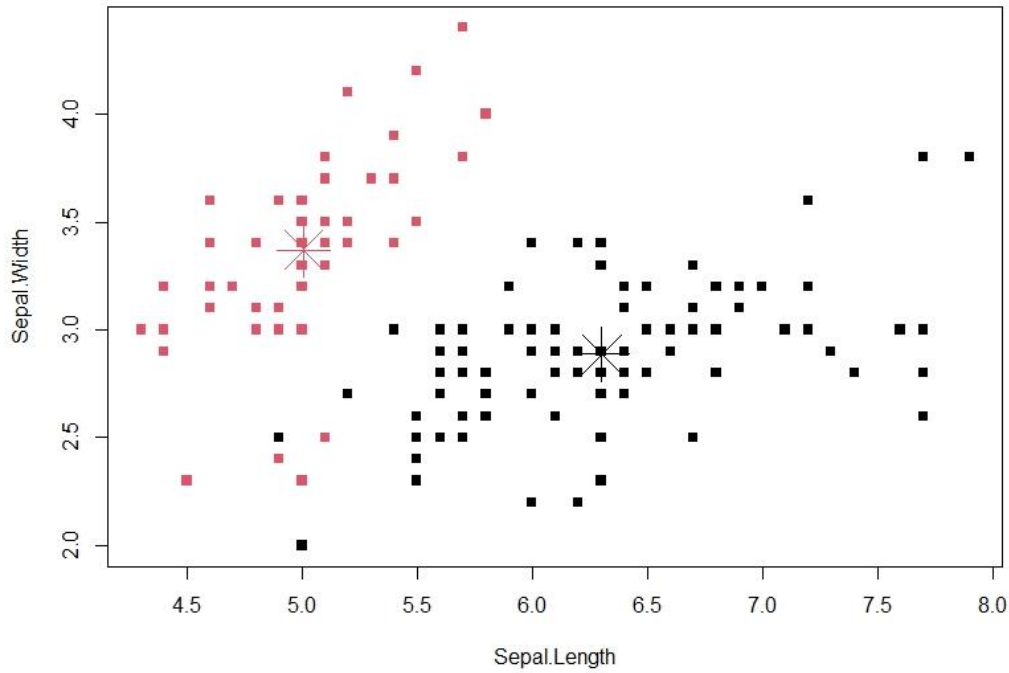
데이터 관측치 수($size)을 확인할 수 있다.

```
               1  2
setosa         0 50
versicolor    47  3
virginica     50  0
```

↳ 기존의 "iris" DataSet와의 교차표를 출력하면 virginica품종의 군집이 사라져 1번 군집으로 분포되어진 것을 볼 수 있다.

(2) K-means Clustering 시각화(2/2)

K-means Clustering을 한 결과의 시각화는 다음과 같다.



3. 결론

각 데이터를 여러 기법을 사용하여 분석하였고 그에 맞는 결과를 얻고 비교를 하였다.

분류 분석을 함에 있어 RandomForest 기법의 성능이 가장 좋았다는 결과를 얻게 되었고,

예측 분석 또한 비교대상이 되는 기법에 비해 상대적으로 RandomForest 기법이 성능이 좋다는 결과를 얻었다.

K-means Clustring을 통하여 군집화를 통해 시각화한 자료로 대상이 되는 데이터 셋의 구성을 직관적으로 파악할 수 있다는 것을 알게 되었다.

4. 부록

사용코드 ; 세부사항 첨부자료 *# 보고서_코드(주).R참고*

```
install.packages('rpart')
install.packages('rpart.plot')
install.packages('neuralnet')  library(rpart)
                    # 의사결정트리 기법
library(rpart.plot) # 의사결정 트리 시각화
library(e1071)   # 나이브베이즈 기법
library(randomForest) # 랜덤포레스트
library(nnet) # 인공신경망
library(caret)
library(car)
library(ggplot2) # 시각화를 위한 패키지
library(neuralnet)  # 인공신경망
#########################################
# 1) 위스콘신 분석
# 데이터 가져오기
wisc <- read.csv('wisc.csv', header = T)
wisc
head(wisc)
str(wisc)
View(wisc)
wisc <- wisc[,-1]
length(wisc)
# 샘플링
set.seed(2) # set.seed를 하지않으면 매번 다른 결과값을 보임.
wi <- sample(1:nrow(wisc), nrow(wisc) * 0.7)
train_wi <- wisc[wi,]
test_wi <- wisc[-wi,]
#  1-1) 의사결정트리 기법   - 정규화 필요X
rpart_wi <- rpart(diagnosis ~ ., data = train_wi)
rpart_wi
```

```
rpart.plot(rpart_wi)
# 예측 범주값 벡터 생성
pred_rpart_wi <- predict(rpart_wi, newdata = test_wi, type = 'class')
# 의사결정 트리 적용 분류 결과 도출
table(test_wi$diagnosis, pred_rpart_wi)
str(pred_rpart_wi)
test_wi$ diagnosis<- as.factor(test_wi$diagnosis)
table(test_wi$diagnosis, pred_rpart_wi)
# 모델 성능 평가 지표
confusionMatrix(pred_rpart_wi, test_wi$diagnosis, positive = 'B')
# confusionMatrix함수를 사용하려면 안의 요소가 factor여야함
# 해석
# 의사결정 트리기법을 적용 시 정확도(101+55)/(101+8+7+55)는 0.9123이다.
# F-Measure 수치는 (2 * 0.9352 * 0.9266) / (0.9352 + 0.9266) = 0.9309 이다.
# 1-2) 나이브베이즈 머신러닝 기법
bayes_wi <- naiveBayes(diagnosis ~ ., data = train_wi)
bayes_wi
# 예측 범주값 벡터 생성
pred_bayes_wi <- predict(bayes_wi, newdata = test_wi, type = 'class')
# 나이브베이즈 적용 분류 결과 도출
table(test_wi$diagnosis, pred_bayes_wi)
# 모델 성능 평가 지표
confusionMatrix(pred_bayes_wi, test_wi$diagnosis)
# 해석
# 나이브베이즈 머신러닝 기법을 적용 시 정확도
(101+60)/(101+3+7+60)는 0.9415이다.
# F-Measure 수치는 (2 * 0.9352 * 0.9712) / (0.9352 + 0.9712) = 0.9527 이다.
# 1-3) 랜덤 포레스트
random_wi <- randomForest(as.factor(diagnosis) ~ ., train_wi, ntree = 500)
# 예측 범주값 벡터 생성
pred_random_wi <- predict(random_wi, test_wi, type = 'response')
# 랜덤 포레스트 적용 분류 결과 도출
table(pred_random_wi, test_wi$diagnosis)
# 모델 성능 평가 지표
confusionMatrix(pred_random_wi,as.factor(test_wi$diagnosis))
```

```
# 해석 : 랜덤 포레스트 적용시 정확도(104+ 61)/(104+ 2+ 4+ 61)는 0.9649이다
# F-Measure 수치는 (2 * 0.9630 * 0.9811) / (0.9630 +  0.9811) = 0.9720 이다.
# 결론
# 따라서 정확도, F-measure으로 비교했을  때,
# 의사결정트리기법 > 나이브 베이즈 > 랜덤 포레스트 기법 순으로 성능이 우수하다
##############################################
# 2) 전복 나이
Abalone <-read.csv('abalone.csv')
str(Abalone)
head(Abalone)
summary(Abalone)
# 컬럼네임
colnames(Abalone) = c('Sex', 'Length', 'Diam', 'Height', 'Whole', 'Shucked', 'Viscera',
'Shell',  'Rings')
str(Abalone)
head(Abalone)
# 샘플링
set.seed(1)
ab <- sample(1:nrow(Abalone), nrow(Abalone) *0.7)
train_ab <- Abalone[ab,]
test_ab <- Abalone[-ab,]
train_ab2 <- Abalone[ab,]
test_ab2 <- Abalone[-ab,]
nrow(test_ab)
nrow(train_ab)
# 2-1) 랜덤포레스트
rf_ab <- randomForest(Rings ~ ., data = train_ab , mtry = 3)
rf_ab
# 시각화
plot(rf_ab, main = '트리 개수 변화에 따른 오류 감소 추이')
# 중요변수확인, 시각화
importance(rf_ab)
varImpPlot(rf_ab, main = '변수 중요도')
# 예측
pre_rf_ab <- predict(rf_ab, newdata = test_ab)
```

```
# 시각화를 위한 처리
df_rf_ab <- data.frame(pre_rf_ab, test_ab$Rings)
head(df_rf_ab)
# 랜포 시각화
ggplot(df_rf_ab, aes(x=pre_rf_ab, y=test_ab.Rings)) +
  geom_point() +  geom_smooth(method = 'auto', se = F) +
  labs(x = '예측 값', y = '실측 값', title = 'Random Forest 예측 결과 비교 산점도')
cor(pre_rf_ab, test_ab$Rings) # 상관관계 분석
RMSE(pre_rf_ab, test_ab$Rings)
mean((pre_rf_ab - test_ab$Rings)^2) # MES평균제곱오차 낮을수록 좋다
# 2-2) 다중회귀분석
# 귀무가설 : rings에 다른 변수들은 영향을 미치지않느다
# 대립가설 : rings에 달느 변수들은 영향을 미친다.
# 회귀분석
lm_Ab <- lm(Rings ~ ., data = train_ab)
summary(lm_Ab)
# 변수선택법을 통한 다중회귀분석, 다중 공선성 확인
lm_Ab2 <- step(lm_Ab, method = 'both')
summary(lm_Ab2)
vif(lm_Ab2)
# 변수 재설정 후 재 회귀
lm_Ab3 <- lm(Rings ~ Sex +  Diam +  Height +  Shucked +  Viscera +  Shell, data
= train_ab)
lm_Ab3
summary(lm_Ab3)
vif(lm_Ab3)
# 변수 재설정 후 재 회귀2
lm_Ab4 <- step(lm_Ab3, method = 'both')
summary(lm_Ab4)
vif(lm_Ab4)
plot(lm_Ab4, which = 1:6)
# 예측
pre_lm_ab <- predict(lm_Ab4, newdata = test_ab)
# 시각화를 위한 처리
df_lm_ab <- data.frame(pre_lm_ab,  test_ab$Rings)
```

```
head(df_lm_ab)
# 회귀 시각화
ggplot(df_lm_ab, aes(x=pre_lm_ab, y=test_ab.Rings)) +
  geom_point() +  geom_smooth(method = 'auto', se = F)+
  labs(x = '예측 값', y = '실측 값', title = '회귀분석 예측 결과 비교 산점도')  # 시각 자료
cor(pre_lm_ab, test_ab$Rings)  # 상관관계 분석
RMSE(pre_lm_ab, test_ab$Rings)
mean((pre_lm_ab - test_ab$Rings)^2) # MES평균제곱오차 낮을수록 좋아
# 2-3) 인공신경망
# 문자를 숫자형으로 변환
train_ab2$Sex[train_ab2$Sex == 'F'] <- '1'
train_ab2$Sex[train_ab2$Sex == 'M'] <- '2'
train_ab2$Sex[train_ab2$Sex == 'I'] <- '3'
test_ab2$Sex[test_ab2$Sex == 'F'] <- '1'
test_ab2$Sex[test_ab2$Sex == 'M'] <- '2'
test_ab2$Sex[test_ab2$Sex == 'I'] <- '3'
train_ab2$Sex <- as.numeric(train_ab2$Sex)
test_ab2$Sex <- as.numeric(test_ab2$Sex)
str(train_ab2)
str(test_ab2)
# 정규화
normal <- function(x){
  return((x - min(x))/(max(x) -  min(x)))
}
train_ab_nor <- as.data.frame(sapply(train_ab2, normal))
test_ab_nor <- as.data.frame(sapply(test_ab2, normal))
str(train_ab_nor)
str(test_ab_nor)
#인공신경망 모델 생성
neur_ab <- neuralnet(Rings ~ ., data = train_ab_nor, hidden = 5, stepmax = 1e+05)
# 인공신경망 시각화
plot(neur_ab)
#예측 결과 생성
```

```
neur_ab_result <- compute(neur_ab, test_ab_nor[1:8])
neur_ab_result$net.result
pre_neur_ab <- predict(neur_ab, test_ab_nor)
str(pred_neur_ab)
# 시각화를 위한 처리
df_neur_ab <- data.frame(pre_neur_ab, test_ab_nor$Rings)
head(df_neur_ab)
# 시각화
ggplot(df_neur_ab, aes(x=pre_neur_ab,  y=test_ab_nor.Rings)) +   geom_point() +
geom_smooth(method = 'auto', se =  F)
cor(pre_neur_ab, test_ab_nor$Rings)
RMSE(pre_neur_ab, test_ab_nor$Rings)
mean((pre_neur_ab -test_ab_nor$Rings)^2)
# 3) K-means시각화
data(iris)
# Species 컬럼 제거
iris2 <- iris[1:4]
set.seed(3)
# 제곱합의 그래프
wssplot <- function(iris2, nc=15, seed=3){
  wss <- (nrow(iris2)-1)*sum(apply(iris2,2,var))
  for (i in 2:nc){
    set.seed(3)
    wss[i] <- sum(kmeans(iris2, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of
squares")}
# 제곱합의 그래프 시각화
wssplot(iris2)
# 군집수의 결정을 위한 패키지
install.packages('NbClust')
library(NbClust)
nc_iris2 <- NbClust(iris2, min.nc = 2, max.nc = 15, method = 'kmeans')
table(nc_iris2$Best.nc[1,])
par(mfrow = c(1,1))
barplot(table(nc_iris2$Best.nc[1,]),
```

```
        xxlab="Numer of Clusters", ylab="Number of Criteria")
# Kmeans 알고리즘 클러스터링 2개 생성
kmeans_result <- kmeans(iris2, 2, nstart = 25)
kmeans_result
kmeans_result$size
kmeans_result$centers
# 실측값과 클러스터링 값 비교
table(iris$Species, kmeans_result$cluster)
# 시각화
plot(iris2[c('Sepal.Length', 'Sepal.Width')], col = kmeans_result$cluster, pch = 15)
points(kmeans_result$centers[,c("Sepal.Length","Sepal.Width")], col = 1:3, pch = 8,
cex = 4)
```

5. 참고자료

○ DataSet 설명

Breast Cancer Wisconsin (Diagnostic) Data Set
https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

Abalone Data Set
https://archive.ics.uci.edu/ml/datasets/Abalone

○ 첨부자료

보고서_코드(주).R
보고서_코드(보조1).R
보고서_코드(보조2).R