머신러닝 기반 데이터 분석

사례연구 4

A3조 남원식, 문태웅, 이순규

목 차

Chapter. 1 - 머신러닝 분류 기법

1) Wisconsin Diagnostic Breast Cancer Data

Wisconsin Diagnostic Breast Cancer Data의 데이터는 32개의 컬럼으로 이루어져있다. 데이터 의 변수는 ID,진단결과,30개의 실측값을 가진다.

각 변수에 대한 설명	
ID	환자 식별번호
diagnosis	양성여부 (M = 악성 ,B = 양성)
각 세포에 대한 정보	
radius	반경
texture	질감
perimeter	둘레
area	면적
smoothness	매끄러움
compactness	조그만정도
concavity	오목함
points	오목한 점의수
symmetry	대칭
dimension	프렉탈 차원
접미어 설명	
_mean	컬럼 3 ~ 12번의 값은 평균값임
_se	컬럼 13~22번의 값은 표준오차임
_worst	나머지컬럼은 구분에서 최댓값의 평균임

2) 나이브 베이즈 분류 기법

나이브 베이즈 기법을 사용해 분류하였다. 데이터 수치의 보존을 위해 seed를 1로 설정하였다.

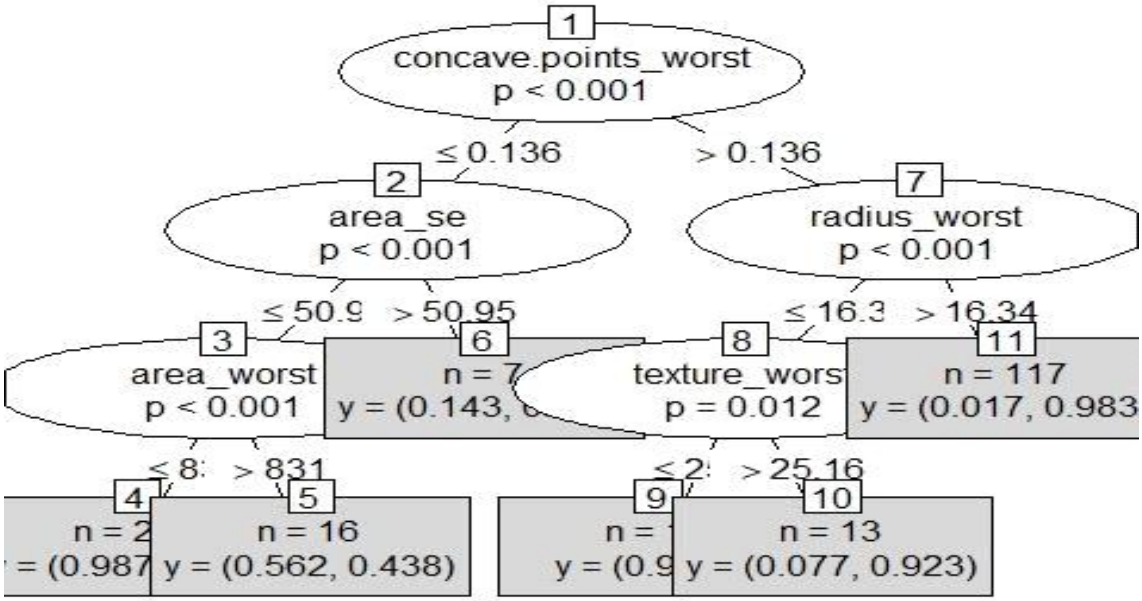
* confusionMatrix를이용하여 예측 결과와 테스트셋

Accuracy	0.9353
Sensitivity	0.9626
Specificity	0.8889

3) 의사 결정 나무 분류 기법

의사 결정 기법을 사용해 분류하였다. 데이터 수치의 보존을 위해 seed를 1로 설정하였다.

*의사 결정 나무 플롯



위의 의사결정나무의 플롯을 보게 되면 뿌리는 오목함의 정도 최댓값의 평균을 기준으로 0.136보다 작으면 왼쪽 크면 오른쪽으로 나뉘게 된다. 2번 노드의 경우는 둘레의 표준오차를 기준으로 50.9보다 작으면 3번노드 50.9보다 크면 6번노드로 나뉘게 된다.3번노드의 경우는 둘레의 최댓값의 평균을 기준 831보다 작으면 4번노드 크면 5번노드로 분류가 됨. 다시 뿌리로 돌아가서 0.136보다 큰 값을 갖는 값들은 7번 노드로 가게 된다. 7번노드는 반경이 16.3보다 작은 경우 8번노드 큰 경우는 11번노드로 분류가 됨. 8번의 노드는 질감의 최댓값의 평균값이 25.16보다 작은 경우 9번 큰 경우는 10번으로 분

류가 된다. 최종적으로 분류가 된 노드들은 4,5,6,9,10,11 총 6개로 분류가 되며 각각의 데이터 분류는 각각 B(양성),M(악성)으로 구분이 된다. 4번의 경우 235개의 개체수와 0.987, 0.013의 분류율을 갖고 5번의 경우 16개의 개체수와 0.562,0.438의 분류율 6번의 경우 7개의 개체수와 0.143,0.857의 분류율 , 9번노드는 10개의 개체수 0.9,0.1의 분류율 10번노드는 13개의 개체수 0.077,0.923의 분류율 11번의 노드는117개의 개체수를 가지며 0.017,0.983의 분류율을 갖는다.

* confusionMatrix를 이용하여 예측 결과와 테스트셋

Accuracy	0.9006
Sensitivity	0.9320
Specificity	0.8529

4) 나이브 베이즈와 의사 결정나무 분류 기법의 결과를 비교
나이브 베이즈 **분류정확도,민감도,특정도는 각각 0.9353 / 0.9626 / 0.8889**
의사 결정 나무 **분류정확도,민감도,특정도는 각각 0.9006 / 0.9320 / 0.8529**
위데이터는 암의 양성과 악성에 대한 비교이므로 악성을 잘못 판단하였을 경우에 문제가 발생할 수 있기 때문에 특정도가 더 중요한 것으로 판단이 된다. 위 두개의 기법을 비교한 결과 나이브 베이즈의 특정도가 더 높게 나올 뿐만 아니라 분류정확도와 민감도 모두 좋은 수치를 가지고 있기 때문에 의사결정 나무보다 Wisconsin Diagnostic Breast Cancer Data를 분류 하는데 더 정확한 기법이라고 판단이 된다.

Chapter. 2 - 머신러닝 예측 기법

1) Abalon Data

Abalon Date의 데이터는 4177개의 데이터와 9개의 컬럼으로 이루어져있다.

각 변수에 대한 설명	
Sex	성별
Length	길이
Diameter	직경
Height	높이
Whole weight	전체 무게
Shucked weight	축소된 무게
Viscera weight	내장 무게
Shell weight	껍질 무게
Rings	종속 변수

2) 선형 회귀 분석 예측 기법

*lm()을 이용한 Abalone 데이터의 선형회귀분석 수치

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	3.89464	0.29157	13.358	< 2e-16	***
SexI	-0.82488	0.10240	-8.056	1.02e-15	***
SexM	0.05772	0.08335	0.692	0.489	
Length	-0.45834	1.80912	-0.253	0.800	
Diameter	11.07510	2.22728	4.972	6.88e-07	***
Height	10.76154	1.53620	7.005	2.86e-12	***
Whole.weight	8.97544	0.72540	12.373	< 2e-16	***
Shucked.weight	-19.78687	0.81735	-24.209	< 2e-16	***
Viscera.weight	-10.58183	1.29375	-8.179	3.76e-16	***
Shell.weight	8.74181	1.12473	7.772	9.64e-15	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

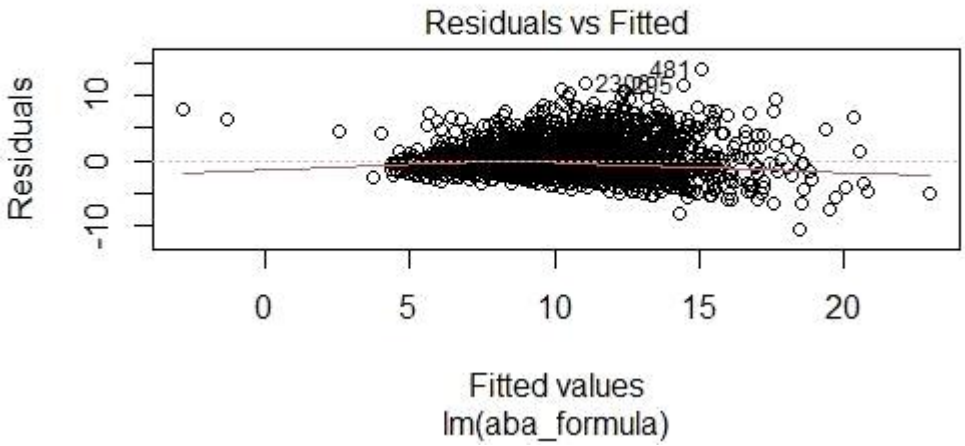
Residual standard error: 2.194 on 4167 degrees of freedom

Multiple R-squared: 0.5379,Adjusted R-squared: 0.5369

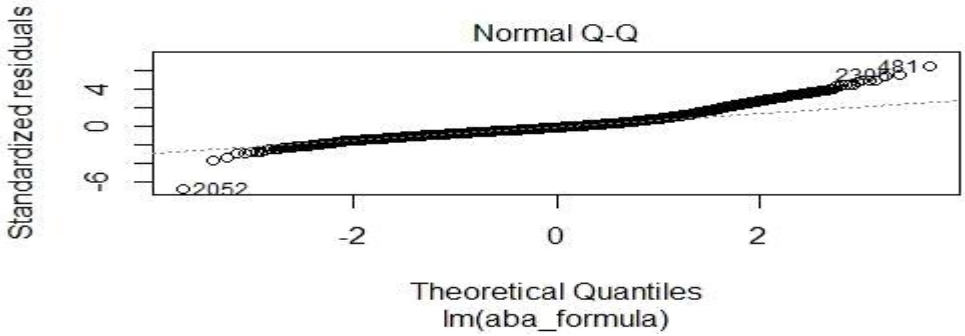
F-statistic: 538.9 on 9 and 4167 DF, p-value: < 2.2e-16

- 검정 결과 R 값은 0.5369, F 통계량 538.9으로 타당성은 양호하지만 계수들 중 일부 유의하지 않은 것으로 분석 된다. 모든 계수가 유의하지는 않지만 타당성이 불량은 아니므로 예측 분석을 시행한다.

* Abalone 데이터 선형회귀분석 플롯 분석



Residual vs Fitted plot은 모델에서 산출된 수치와 잔차를 분석한 결과이다. 잔차가 0값을 중심으로 퍼져 있는 축에 속하는 것으로 보여진다. 별다른 트렌드를 보여주지는 않는다.

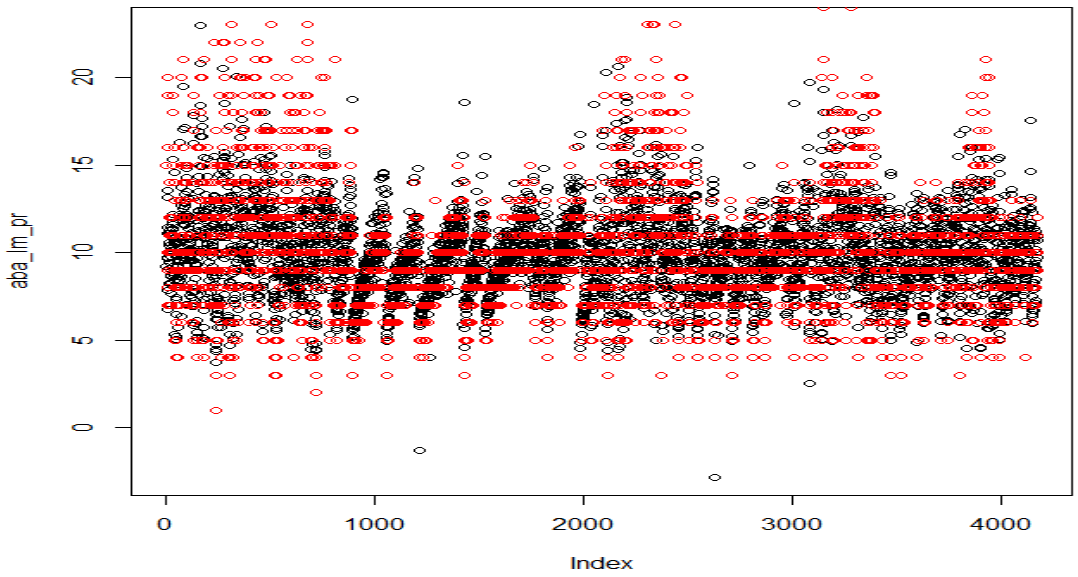


Nomal Q~Q플롯은 정규성 분표를 확인할 수 있는 plot으로 45도 선에 고르게 분포하면 이상적이라 볼 수 있으나 좌 하향 우상향 된 기울기를 그리며 그래프를 보여준다.

상관계수를 통한 예측 결과

정확도 : 0.7101704

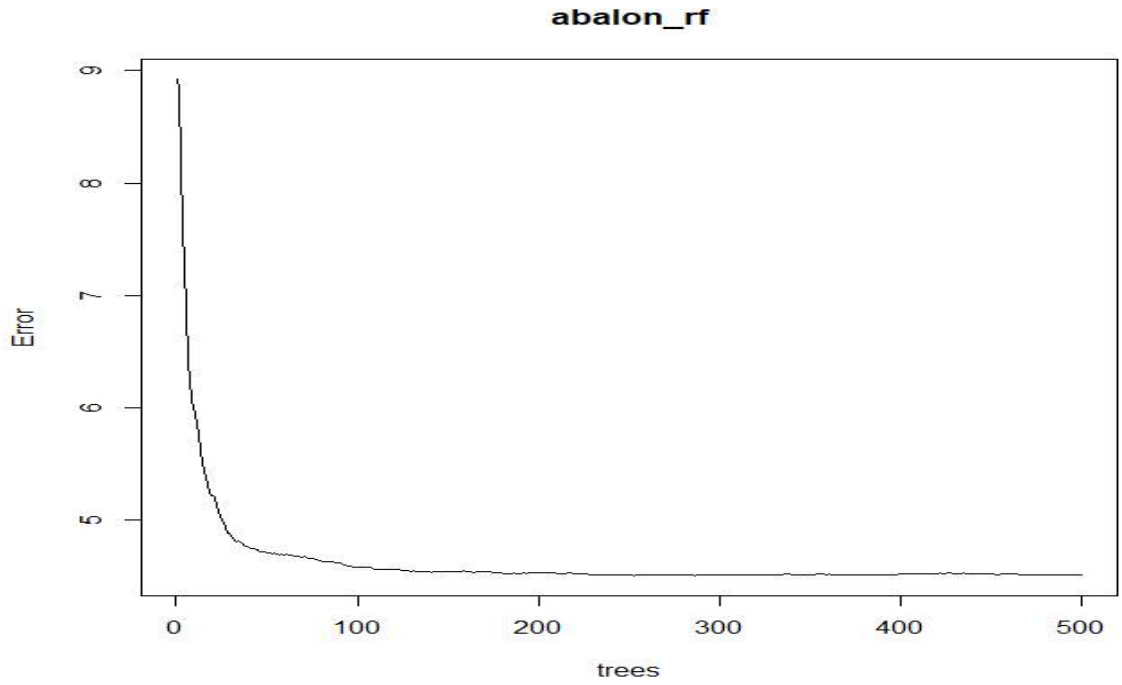
※ 예측값과 실제값의 비교 (black – 예측값, red – 실제값)



3) 랜덤 포레스트 예측 기법

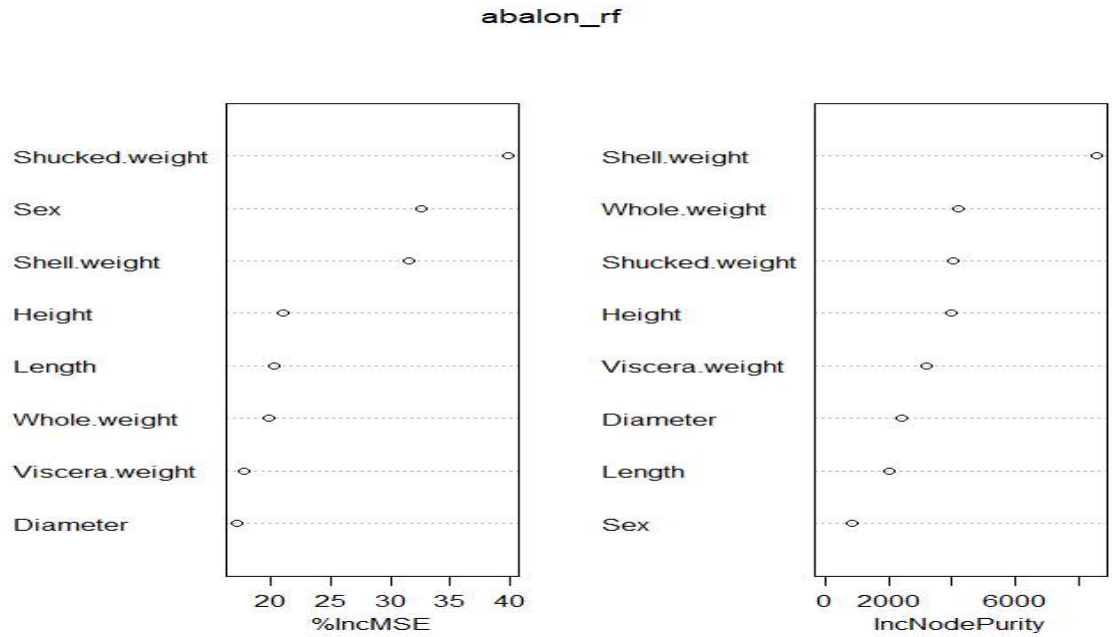
랜덤포레스트는 기존 의사결정트리와 앙상블의 배깅 알고리즘을 개선하여 더 좋은 성능을 내는 머신러닝 예측 방법이다. 하지만 하이퍼파라미터 지정에 따른 성능의 변화가 심하여 파라미터 지정에 신경 써줘야 한다.

1. Mtry(트리의 노드 결정시 독립변수의 갯수) : 3
2. 연산에 참가할 트리 생성 수 : 500



랜덤 포레스트 시각화(트리 개수 변화에 따른 오류 감소 추이)

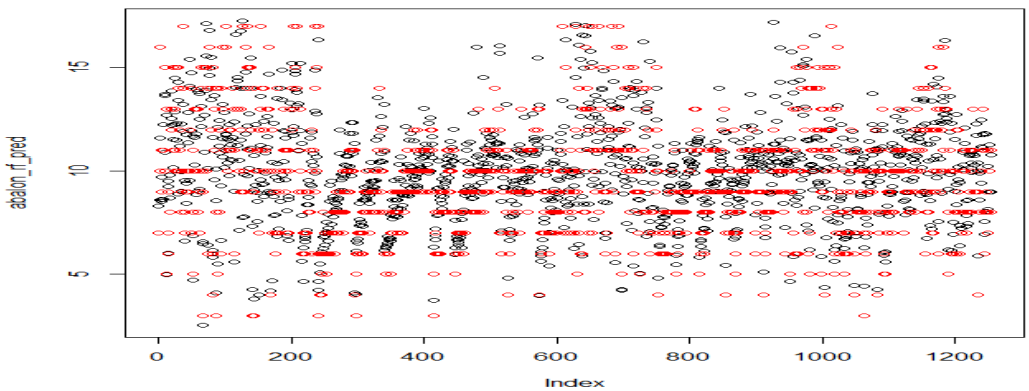
변수 중요도 시각화



랜덤 포레스트 분석시 사용된 변수의 중요도는 위와 같다.
Shucked(축소된 무게) 변수와 Shell(껍질무게) 두 변수가 가장 중요한 변수이고, 두 번째 변수는 Sex(성별)과 Whole(전체무게)이다.
이 상위 두가지 변수가 분석에 압도적인 중요성을 보여주고 있으며, 성별 변수를 제외한다면 무게를 가진 변수들이 중요성을 보여주고 있다 볼 수 있다.

상관계수를 통한 예측 결과
정확도 : 0.730932

예측값과 실제값의 비교 (black – 예측값, red – 실제값)



4) 선형 회귀 분석과 랜덤 포레스트 예측 기법의 결과를 비교
선형 회귀 분석의 **정확도는 0.7101704 / 약 71%**
랜덤 포레스트의 **정확도는 0.730932 / 약 73%**

예측 성능은 내림차순으로 다음과 같다.
랜덤 포레스트 기법 → 선형 회귀 분석 으로 정확도를 비교하여 볼 수 있다.

하지만, 선형회귀 분석의 경우 변수들의 독립성은 보장되어 지나 정규성, 선형성에 있어 유의확률을 만족 시킬 수 없었기 때문에 낮은 예측 정확도를 확인할 수 있었으며, 랜덤포레스트의 경우 높은 예측 정확도를 보여주었지만 파라미터 지정에 예민하며, 그렇기 때문에 미세한 파라미터 조정으로 성능 발전을 기대할 수 있다.

Chapter. 3 – 군집 분석

1) K-means clustering 기법

각 군집에 할당된 포인트의 평균 좌표를 보고 군집의 중심점을 업데이트 시키는 모델로 거리의 개념이 없어지는 고차원의 경우에는 제대로 된 성능이 나오지 않는다는 단점이 있습니다.

*kmeans() 이용한 iris 데이터의 clustering

```
> kmeans_result <- kmeans(iris2, 2)
> kmeans_result
K-means clustering with 2 clusters of sizes 53, 97

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.005660    3.369811     1.560377    0.290566
2     6.301031    2.886598     4.958763    1.695876

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
 [52] 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2
[103] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1]  28.55208 123.79588
 (between_SS / total_SS =  77.6 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
```

iris data에서 Species 열을 제외시킨 연속변수만 분석에 포함하여 2개의 클러스터로 나뉘어 실행 이렇게 나뉘어진 클러스터에는 각각 53개, 97개의 데이터를 가지고 있는 것이 확인됨 또한 **분류 정확도는 77.6%**임을 확인할 수 있다.

*table() 함수를 이용하여 군집 라벨을 실제 iris 종(species) 라벨과 비교

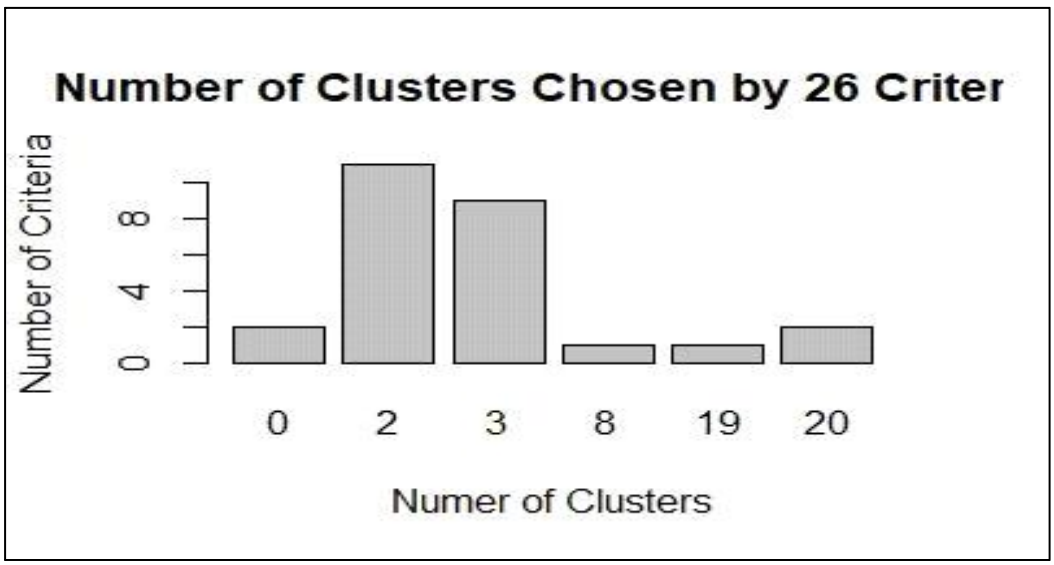
```
> table(iris$Species,kmeans_result$cluster)

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

위 테이블에서 행은 실제값, 열은 K-Means로 클러스터링된 예측값으로 볼 수 있다. 참고로 1이 반드시 setosa를 의미한다고는 볼 수 없으며, 해석하자면 setosa는 1번 클러스터와 비슷한 특징을 지니고 있다 정도로 볼 수 있다.

*NbClust 함수를 사용하여 군집수의 신뢰성 확인
1) 최적의 군집수 : 2

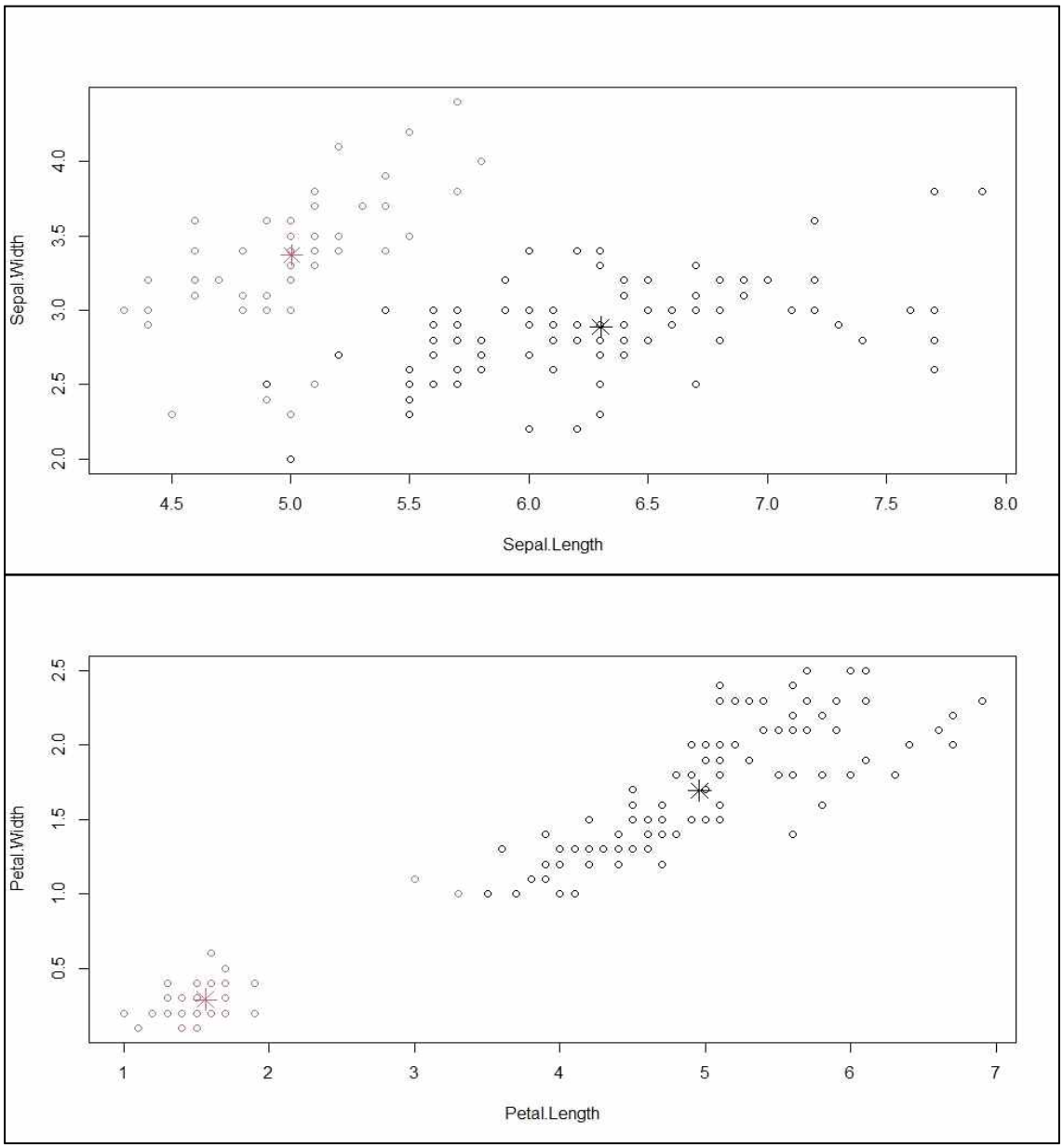
```
> library(NbClust)
> set.seed(1)
> iris_c <- NbClust(iris2, min.nc=2, max.nc=15, method="kmeans")
*** : The Hubert index is a graphical method of determining the number of clusters.
                In the plot of Hubert index, we seek a significant knee that corresponds to a
                significant increase of the value of the measure i.e the significant peak in Hubert
                index second differences plot.

*** : The D index is a graphical method of determining the number of clusters.
                In the plot of D index, we seek a significant knee (the significant peak in Dindex
                second differences plot) that corresponds to a significant increase of the value of
                the measure.

*******************************************************************
* Among all indices:
* 11 proposed 2 as the best number of clusters
* 11 proposed 3 as the best number of clusters
* 1 proposed 8 as the best number of clusters
* 1 proposed 12 as the best number of clusters

                   ***** Conclusion *****

* According to the majority rule, the best number of clusters is  2
```

2) NrClust 시각화



NbClust()를 이용하여 값을 구하고 그래프화 한 결과 **최적의 군집수는 2**라는 것을 알 수 있고 군집수를 2개로 설정하여 그래프를 생성하였다.

2) 시각화



Chapter. 4 결론

각각의 데이터 셋에 대한 분류와, 예측기법을 활용한 보고서이다.

분석기법

wiscon 유방암 데이터셋에서 diagnosis을 종속변수로 설정하고 나머지 변수를 독립변수로 설정하여 각각의 분석 기법을 실시하였을 때, 어느 분석기법이 더 우수한지 비교 분석 할 수 있는 결과를 도출 했다.

예측기법

abalon 전복 데이터셋에서 Rings을 종속변수로 설정하고 나머지 변수를 독립변수로 설정하여 각각의 예측 기법을 실시하였을 때, 어느 예측기법이 더 우수한지 비교 분석 할 수 있는 결과를 도출 했다.

Chapter. 5 사용 R코드 및 참고자료

분석기법 (사용 R코드)

```
#### 1-1 나이브 베이즈 모형 ####
rm(list = ls())
install.packages("e1071")
install.packages("caret")
library(e1071)
library(caret)

setwd("C:/Rwork/data")
wisc <- read.csv("wisc_bc_data.csv")
#불필요 NA값만들어있는 컬럼 제거
wisc <- subset(wisc, select = -X)
wisc
#요인형으로 변환
wiscf <- factor(wisc$diagnosis)
wisc$diagnosis2 <- wiscf
wisc$diagnosis <- NULL
#포뮬러
wisc_formula <- diagnosis2 ~ + radius_mean + texture_mean + perimeter_mean +
  area_mean + smoothness_mean + compactness_mean + concavity_mean + concave.points_mean +
  symmetry_mean + fractal_dimension_mean + radius_se + texture_se + perimeter_se +
  area_se + smoothness_se + compactness_se + concavity_se + concave.points_se +
  symmetry_se + fractal_dimension_se + radius_worst + texture_worst + perimeter_worst +
  area_worst + smoothness_worst + compactness_worst + concavity_worst + concave.points_worst +
  symmetry_worst + fractal_dimension_worst
# 트레이닝, 테스트셋
set.seed(1)
idx <- createDataPartition(y=wisc$diagnosis,p=0.7,list=FALSE)
train_wisc <- wisc[idx,]
test_wisc <-  wisc[-idx,]
# 베이지안
bayes_wisc_tr <- naiveBayes(diagnosis2~.,data = train_wisc)
# 테스트셋 검정
pre_bayes_wisc <- predict(bayes_wisc_tr,test_wisc,type = "class")
table(pre_bayes_wisc,test_wisc$diagnosis)
```

```r
bayes_wisc_fin <- as.factor(test_wisc$diagnosis)
confusionMatrix(pre_bayes_wisc,bayes_wisc_fin)

#### 1-2 의사결정 나무 분류기법 ####
rm(list = ls())
install.packages("party")
install.packages("caret")
library(party)
library(caret)
table(wisc$diagnosis2)

setwd("C:/Rwork/data")
wisc <- read.csv("wisc_bc_data.csv")
#불필요 NA값만들어있는 컬럼 제거
wisc <- subset(wisc, select = -X)
wisc
table(wisc$diagnosis)
str(wisc)
#요인형으로 변환
wiscf <- factor(wisc$diagnosis)
wisc$diagnosis2 <- wiscf
wisc$diagnosis <- NULL
# 트레이닝, 테스트셋
set.seed(1)
idx <- sample(1:nrow(wisc),nrow(wisc)*0.7)
train_wisc <- wisc[idx,]
test_wisc <-  wisc[-idx,]
#포물러 생성
wisc_formula <- diagnosis2 ~ + radius_mean + texture_mean + perimeter_mean +
  area_mean + smoothness_mean + compactness_mean + concavity_mean + concave.points_mean +
  symmetry_mean + fractal_dimension_mean + radius_se + texture_se + perimeter_se +
  area_se + smoothness_se + compactness_se + concavity_se + concave.points_se +
  symmetry_se + fractal_dimension_se + radius_worst + texture_worst + perimeter_worst +
  area_worst + smoothness_worst + compactness_worst + concavity_worst + concave.points_worst +
  symmetry_worst + fractal_dimension_worst
# 의사 사 결정 나무 형성
wisc_tree1 <- ctree(wisc_formula, data = train_wisc)
wisc_tree1
plot(wisc_tree1,type = "simple")
# 테스트셋 검정 오분류율 테스트
predicted_wisc <- predict(wisc_tree1,test_wisc)
plot(predicted_wisc)
text(predicted_wisc)
confusionMatrix(predicted_wisc,test_wisc$diagnosis2)
```

예측기법 (사용 R코드)

```
#### 2-1 선형 회귀 예측 기법 ####
library(car)
abalon_lm <- lm(Rings ~ ., data = aba_train)
summary(abalon_lm)

abalon_lm_res <- residuals(abalon_lm)
durbinWatsonTest(abalon_lm_res)
# 1.996 이므로 독립선 상한인 1.69보다 크므로 계수들은 서로 독립적이다 볼 수 있음.

windows()
par(mfrow = c(1,1))
plot(abalon_lm)  # 등분산성 확인

shapiro.test(abalon_lm_res)

# 예측 및 결과
abalon_data_pred <- predict(abalon_lm, newdata = aba_test)
cor(abalon_data_pred, aba_test$Rings) # 상관계수 결과 0.7101704 / 71%
windows()
plot(abalon_data_pred)
points(aba_test$Rings, col = "red")

#### 2-2 랜덤 포레스트 예측 기법 ####
rm(list = ls())
getwd()
setwd()
abalone_data <- read.csv('abalone.csv', header = T)
head(abalone_data); str(abalone_data) #데이터 확인

# 학습데이터, 검정데이터 셋으로 분리
set.seed(1) # 매번 값을 변동 안시키기 위해 설정
idx2 <- sample(1:nrow(abalone_data), 0.7*nrow(abalone_data))
aba_train <- abalone_data[idx2,]
aba_test <- abalone_data[-idx2,]
```

```r
aba_train$Sex <- as.factor(aba_train$Sex) # 캐릭터값에서 변경
aba_test$Sex <- as.factor(aba_test$Sex)
str(aba_train) # 9개 변수중 종속변수를 제외하면 8개
sqrt(8) # 2.8개이므로 변수는 3개로 사용

abalon_rf <- randomForest(Rings ~ ., data = aba_train, mtry = 3, importance = T)
windows()
plot(abalon_rf)  # 트리 개수 변화에 따른 오류 감소 추이

# 변수 중요도
importance(abalon_rf)  # 변수 중요도
windows()
varImpPlot(abalon_rf)  # 변수 중요도 시각화

# 예측 및 결과
abalon_rf_pred <- predict(abalon_rf, newdata = aba_test)
cor(abalon_rf_pred, aba_test$Rings) # 73% 예측값.
# 상관계수을 이용한 검정 결과 : 0.730932
windows()
plot(abalon_rf_pred)
points(aba_test$Rings, col = "red")
```

군집분석 (사용 R코드)

```r
#### 3. iris데이터에서 species 컬럼 데이터를 제거한 후 k-means clustering를 실행하고 시각화하시오.####
library(ggplot2)
install.packages("mclust")
install.packages("corrgram")
install.packages("NbClust")
library(mclust)
library(corrgram)

data(iris)
head(iris)
str(iris)
iris2 <- iris

# 최적의 군집수 찾기
library(NbClust)
set.seed(1)
iris_c <- NbClust(iris2, min.nc=2, max.nc=15, method="kmeans")
table(iris_c$Best.n[1,])
# 2~ 3개의 군집수가 적당

# species 데이터 제거
iris2$Species <- NULL
head(iris2)

# k-means clustering 실행
kmeans_result <- kmeans(iris2, 6)
kmeans_result
str(kmeans_result)

# 시각화
plot(iris2[c("Sepal.Length", "Sepal.Width")], col=kmeans_result$cluster)
points(kmeans_result$centers[, c("Sepal.Length", "Sepal.Width")], col=1:4, pch=8, cex=2)
plot(iris2[c("Petal.Length", "Petal.Width")], col=kmeans_result$cluster)
points(kmeans_result$centers[, c("Petal.Length", "Petal.Width")], col=1:4, pch=8, cex=2)
```

참고자료

첨부자료

사례연구 4 (통합본).R
사례연구 4 (보조1).R
사례연구 4 (보조2).R
사례연구 4 (보조3).R

wisc_bc_data.csv
abalon_data.csv

참고자료
https://www.kaggle.com/uciml/breast-cancer-wisconsin-data : 위스콘데이터
https://archive.ics.uci.edu/ml/datasets/Abalone : 전복데이터