최종작성: 21.11.21(일)

사례연구 4

위스콘신 주 유방암 데이터와

전복 데이터 셋에 대한 분류/예측

/

붓꽃 데이터 군집분석

C3 조: 오준서 천성한 한호종 황윤수

21.11.22(월) 제출

목차

서론 3

본론 4

1. Wisconsin Breast Cancer Data Sets 4

1) 데이터 셋 요약 4

2) 나이브 베이즈 기법 5

3) 의사결정트리 기법 6

4) 인공신경망 기법 7

5) 랜덤포레스트 기법 8

6) 기법 비교 9

2. Abalone Data Sets 10

1) 데이터 셋 요약 10

2) 다중회귀분석 기법 11

3) 의사결정트리 기법 13

4) 인공신경망 기법 14

5) 랜덤포레스트 기법 15

6) 기법 비교 16

3. Iris Data Sets 17

1) 데이터 셋 요약 17

2) K-means clustering 기법 적용 17

3) 시각화 18

결론 19

부록 20

참조 32

서론

본 문서는 각종 데이터 셋에 대하여 분류/예측기법을 적용하고 인사이트를 제공하는 보고서이다.

- 위스콘신 주 유방암 관련 데이터를 활용하여 분류기법을 적용하고 양성여부를 판별한다.
- 전복과 관련된 데이터를 활용하여 예측기법을 적용하고 Rings 의 수치를 예측한다.
- R 내장 데이터 셋인 Iris 를 활용하여 K-means clustering 기법을 적용하여 군집분석을 실시한다.

본론

1. Wisconsin Breast Cancer Data Sets

1) 데이터 셋 요약

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667
84300903	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999
84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744
84358402	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613

...(생략) (표 1-1)

아래는 각 컬럼별 설명이다.

: (3)번~(12)번은 세포핵마다 해당특징들이 계산된 값의 평균을 의미한다.

(1) ID: 환자식별번호

(2) Diagnosis (M = malignant, B = benign) : 양성여부 M = 악성, B 양성

(3) radius: 반경(중심에서 외벽까지 거리들의 평균값)

(4) texture : 질감(Gray-scale 값들의 표준편차)

(5) perimeter: 둘레

(6) area: 면적

(7) smoothness: 매끄러움(반경길의 국소적 변화)

(8) compactness (perimeter^2 / area - 1.0) : 조그만 정도(둘레^2/면적-1)

(9) concavity: 오목함(윤곽의 오목한 부분의 정도)

(10) concavePoints: 오목한 점의 수

(11) symmetry: 대칭

(12) fractalDimension ("coastline approximation" - 1) : 프랙탈 차원(해안선근사 - 1)

(3~12) 평균값

(생략)(13~22) 3~12 까지의 순서대로 표준오차

(생략)(23~32) 3~12 까지의 순서대로 각 세포별 구분들의 제일 큰 값 3 개의 평균값

2) 나이브 베이즈 기법

예측결과)

Naive Bayes Classifier for Discrete Predictors

Call:

naiveBayes.default(x = X, y = Y, laplace = laplace)

> # 2-2) 나이브 베이즈 모델 생성 (부록 참조)

> # 2-3) 예측값 생성

Confusion Matrix and Statistics

Reference

Prediction B M

B 101 9

M 6 54

Accuracy : 0.9118

95% CI : (0.8586, 0.9498)

No Information Rate : 0.6294

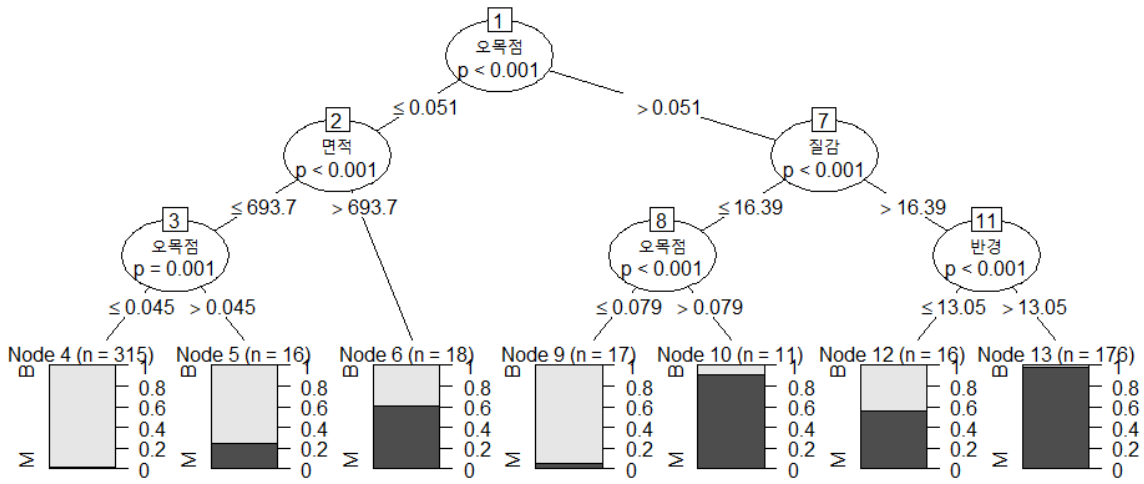
P-Value [Acc > NIR] : <2e-16

Kappa : 0.809

나이브 베이즈를 활용한 혼동행렬 생성결과 예측 정확도는 **91.18%**로 측정되었다.

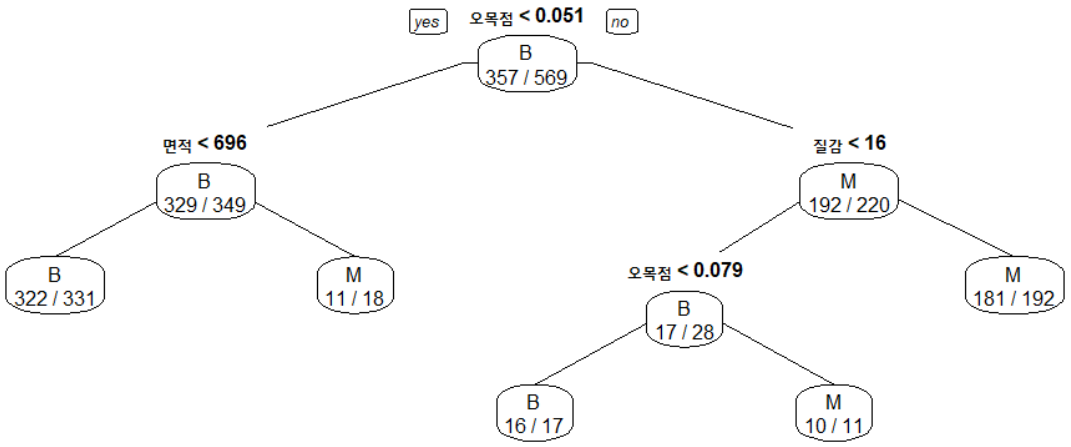
3) 의사결정트리 기법

예측결과)



(그림 1-1)

의사결정트리를 활용하여 분류한 결과 오목점의 개수(concavePoints)가 가장 중요한 변수로 선정 되었다. 위 그림을 조금 더 간결하게 표현하면 다음과 같다.



(그림 1-2)

해석)

첫번째 분류 지점을 보면 총 569 개의 관측치 중 357 개의 관측치가 양성으로 판정되었다. 여기서, **가장 중요한 분류기준은 환자의 세포핵에 평균적으로 오목한 점의 수**이다. 이 점의 수가 **평균 0.051 개 이하이면 62%확률로 양성**이고, **0.051 개 이상이면 38%확률로 악성**으로 판단한다.

최종적으로, 의사결정트리 기법을 실시했을 때 **오목점의 수가 0.051 이상이고 질감이 16 이상일경우 94%확률로 악성**이고, **오목점의 수가 0.051 이하이면서 면적이 696 이하일 경우 97%확률로 양성**이라는 유의미한 확률을 구할 수 있다.

4) 인공신경망 기법

예측결과)

```
>    # 모델정확도 평가
> (1)번모델
 0.8275186
> (2)번모델
 0.7427824
> (3)번모델
 0.7969709
```

Wisconsin Breast Cancer Data 에서 Diagnosis(양성여부)변수를 종속변수로 설정하고 인공신경망으로 분류한 결과 위와 같은 정확도를 가진 모델을 생성했다.

(시각화 자료 및 코드는 (부록 1-4) 참조)

해석)

인공신경망을 통한 분석은 모델을 생성해 예측한 후 정확도는 측정할 수 있으나 시각화를 통한 각 모델에 대한 설명이 어려우므로 모델의 예측정확도만 작성하였다.

(1)번모델: 은닉층 1 개, 정확도 82%

(2)번모델: 은닉층 3 개, 정확도 74%

(3)번모델: 은닉층 4 개, 정확도 79%

테스트 데이터셋*을 대입하여 인공신경망을 통해 예측한 결과는 위와 같다.

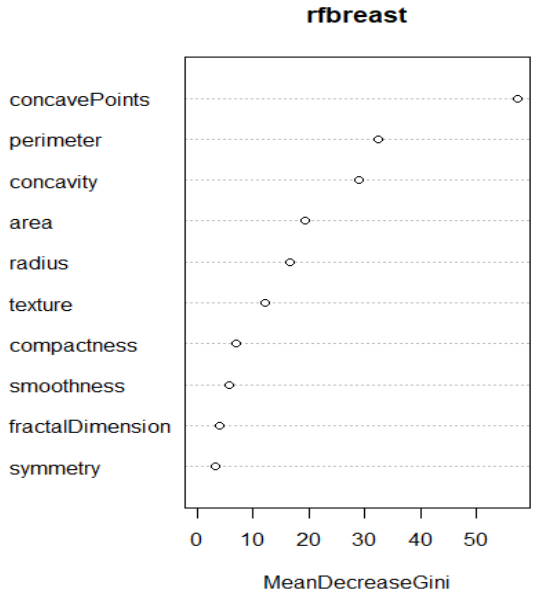
*테스트 데이터셋=원본데이터의 30% 샘플

5) 랜덤포레스트 기법

예측결과)

rf.predbreast	**B**	**M**
B	**103**	**5**
M	**4**	**58**

[1] 0.9470588



(그림 1-3)

해석)

랜덤포레스트 모델은 170 개의 양성여부 관측치 중 161 개의 양성여부를 정확히 판별하여 **94.7% 정분류율**을 보였다.

그림(1-?)는 위에서부터 차례대로 가장 중요한 변수들을 나타낸 그래프이다.

해당 모델은 **양성여부 판별에 concavePoints 를 가장 중요한 변수로 분류**하였다.

코드는 (부록 1-5)참조

6) 기법 비교

Wisconsin Breast Cancer Data 셋에 적용한 분류기법은 아래와 같다.
(나이브 베이즈, 의사결정트리, 인공신경망, 랜덤포레스트)

나이브 베이즈: 다른기법에 비해 쉽게 분류예측을 할 수 있었고 정확도 또한 90% 이상으로 준수한 성능을 보였다.

의사결정트리: 분류의 기준이 명확하여 분류예측 사유에 대한 설명이 쉬웠다.

인공신경망: 은닉층의 개수 설정 및 분류예측 사유 설명이 어려웠다. 또한, ID 와 같은 무의미한 변수가 속해 있을 경우에도 결과값에 영향을 미쳤다.

랜덤포레스트: 무의미한 변수에도 영향을 받지 않았고 다른 모델들에 비해 여러 번 실행해도 높은 정분류율을 보였다.

2. Abalone Data Sets

1) 데이터 셋 요약

Sex	Length	Diameter	Height	WholeWeight	ShuckedWeight	ViscerWeight	ShellWeight	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

...(생략) (표 2-1)

아래는 각 컬럼별 설명이다.

(1) sex = 성별(M=수컷,F=암컷,I=덜자란 개체)

(2) Length = 껍질을 포함한 가장 긴 길이

(3) Diameter = 길이에서 수직으로 껍질 포함한 지름

(4) Height = 전복 몸통을 포함한 높이

(5) wholeWeight = 전체무게

(6) shuckedWeight = 전복 살 무게

(7) visceraWeight = 전복 피를 뺀 내장 무게

(8) shellWeight = 전복의 말린 껍질 무게 / Rings = 껍질의 링 수 / 링 수 + 1.5 = 전복 나이

단, 여기서 wholeWeight > shuckedWeight + visceraWeight + shellWeight 인데
전체 무게를 측정하고 살과 내장, 껍질을 분리 후 측정해서 측정 과정에서 손실된 피와
수분때문으로 보임

전복은 첫해를 제외하고 매년 껍데기에 하나의 고리를 추가하는데, 우리는 고리의 수에
1.5 를 더하면 전복의 나이를 아주 정확하게 추정할 수 있다.

2) 다중회귀분석 기법

예측결과)

Rings 를 **종속변수**로 설정하고 **나머지 모든 변수**를 **독립변수**로 넣어 회귀분석을 실시한 결과이다.

#2_1 다중회귀분석

Call:

lm(formula = formula2, data = train2)

Residuals:

Min	1Q	Median	3Q	Max
-8.1558	-1.3119	-0.3353	0.8698	11.8512

(표 2-2)

Residual standard error: 2.172 on 2914 degrees of freedom

Multiple R-squared: 0.5523, Adjusted R-squared: 0.5511

F-statistic: 449.4 on 8 and 2914 DF, p-value: < 2.2e-16

cor(pred1, test2$Rings) #회귀모델 평가

[1] 0.6942753

Coefficients:	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.7698	0.3202	8.651	< 2e-16
Sex	0.3316	0.0596	5.564	2.87e-08
Length	0.6289	2.1090	0.298	0.76557
Diameter continuous	8.3821	2.6117	3.209	0.00134
Height	19.5893	2.5482	7.688	2.03e-14
Whole weight	9.7050	0.8699	11.157	< 2e-16
Shucked weight	20.8512	-0.9799	-21.279	< 2e-16
Viscera weight	-10.8941	1.5622	-6.973	3.81e-12
Shell weight	7.5864	1.3276	5.715	1.21e-08

(표 2-2)

해석)

이 모델에서 F-통계량=449.4, p-value=2.2e-16 이므로

Rings 에 대한 독립변수들 간의 모형은 유의수준 5%하에서 통계적으로 매우 유의하다

이 모델에서 회귀분석 결과로 추정된 회귀식은

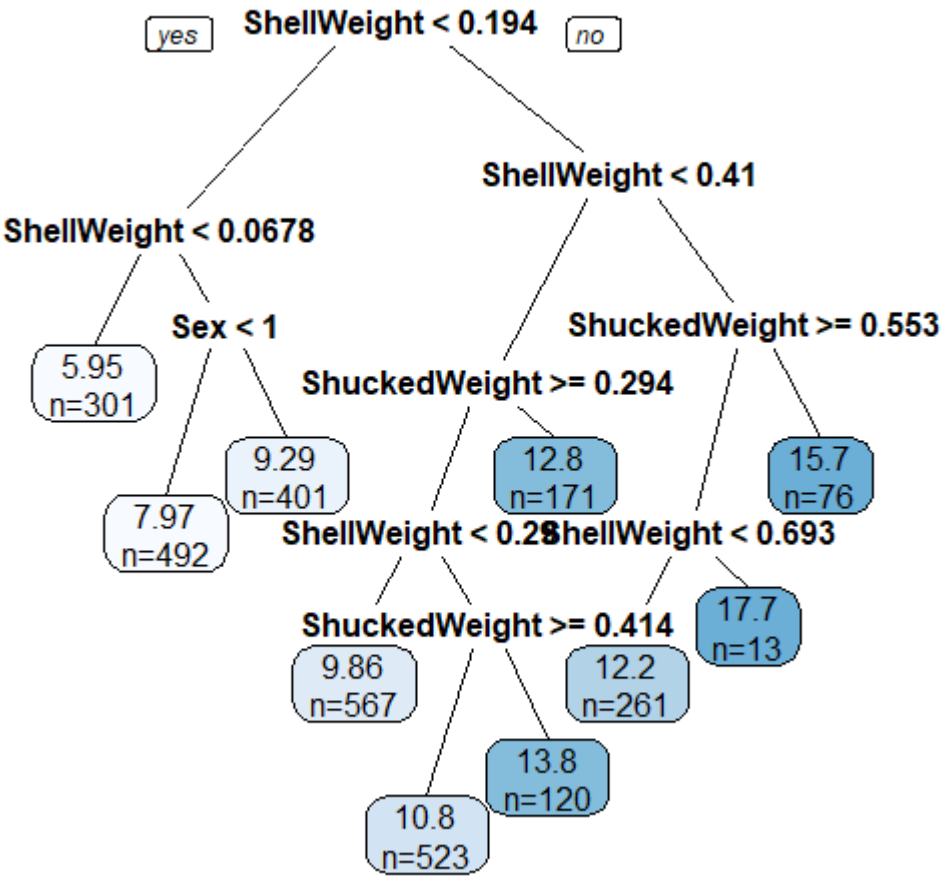
[Rings = 2.7698 + 0.3316(Sex) + 0.6289(Length) + 8.3821(Diameter) + 19.5893(Height) + 9.7050(Whole Weight) + 20.8512(Shucked Weight) – 10.8941(Viscera Weight) + 7.5864(Shell Weight]이다.

여기서, Rings 에 **가장 큰 영향을 미치는 긍정**요인은 **Shucked Weight** 이고, **부정**요인은 **Viscera Weight** 뿐임을 알 수 있다.

따라서, Shucked Weight 가 1 만큼 증가할수록 Rings 는 20.8512 만큼 증가하는 것을 알 수 있다.

3) 의사결정트리 기법

예측결과)



(그림 2-1)

해석)

이 모델에서 **가장 중요한 분류기준은** **Shell Weight**이다. 첫번째 분류 지점 데이터를 계산해보면 **Shell Weight 가 0.194 미만일 때** 총 2,925 개의 관측치 중 1,194 개의 관측치가 **Rings 는 5.95~9.29 사이일 것으로 예측**했다.

4) 인공신경망 기법

예측결과)

```
>    # 상관관계를 통한 정확도 평가
> cor(1 번모델, 테스트 데이터)
 0.5611581
> cor(2 번모델, 테스트 데이터)
 0.6471513
```

Abalone Data 에서 Rings 변수를 종속변수로 설정하고 인공신경망으로 예측한 결과 위와 같은 정확도를 가진 모델을 생성했다.

(시각화 자료 및 코드는 (부록 2-4) 참조)

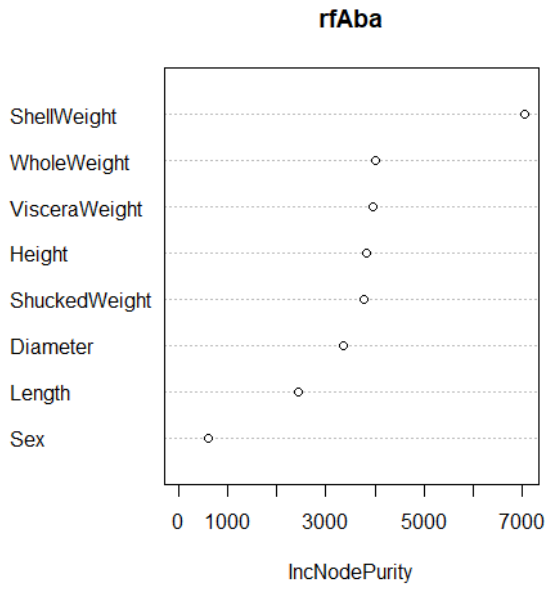
해석)

Abalone Data 에서 인공신경망을 통한 분석결과 **은닉층을 1 개**로 설정했을 때는 **56%의 정확도**를 보였고 **은닉층을 3 개로 설정했을 때**는 **64%의 정확도**로 조금 향상된 모습을 보이긴 하였으나 큰 차이는 없었다.

5) 랜덤포레스트 기법

예측결과)

```
> importance(rfAba)
                IncNodePurity
Sex                609.9672
Length            2439.3227
Diameter          3364.8894
Height            3821.0685
WholeWeight       4006.4064
ShuckedWeight     3765.8801
VisceraWeight     3947.1092
ShellWeight       7054.5213
>   cor(rf.predAba, abaloneTest$Rings)
[1] 0.7214904
```



(그림 2-5)

해석)

랜덤포레스트 예측결과(그림 2-5) 다른 기법들과 마찬가지로 Shell Weight 가 가장 중요한 변수로 측정되었고 정확도는 72%가 나왔다.

6) 기법 비교

다중회귀분석: 종속변수에 미치는 영향을 쉽게 설명할 수 있고 각 독립변수 간에 얼마만큼의 영향을 미치는지 쉽게 파악할 수 있었다. 그러나 무의미한 변수를 걸러내지 않으면 잘못된 결과를 출력할 수도 있다.

의사결정나무: 무의미한 변수가 존재해도 쉽게 걸러낼 수 있다는 것이 장점이었다. 그러나, 가지가 3 개이상 넘어가면 설명하기가 어려워졌다.

인공신경망: 다른 기법의 모델들보다 정확도가 다소 떨어지는 모습을 볼 수 있었다. 또한, 여러 번 돌릴수록 정확도의 변동폭이 컸으며 실행시간이 오래걸리는 등의 단점을 보였다

랜덤포레스트: 가장 영향력이 큰 변수를 찾아내기가 수월했고 무의미한 변수가 있어도 크게 영향을 받지 않았다. 정확도 또한 준수했으며 실행시간도 길지 않았다.

3. Iris Data Sets

1) 데이터 셋 요약

iris 내장 데이터 셋 내용

Index No.	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa

...(생략)(표 3-1)

아래는 각 컬럼 별 설명이다.

Index No: 데이터의 순서 넘버링

Sepal Length: 꽃받침의 길이 정보

Sepal Width: 꽃받침의 너비 정보

Petal Length: 꽃잎의 길이 정보

Petal Width: 꽃잎의 너비 정보

Species: 꽃의 종류 정보(setosa/versicolor/virginica 의 3 종류로 구분)

2) K-means clustering 기법 적용

적용결과)

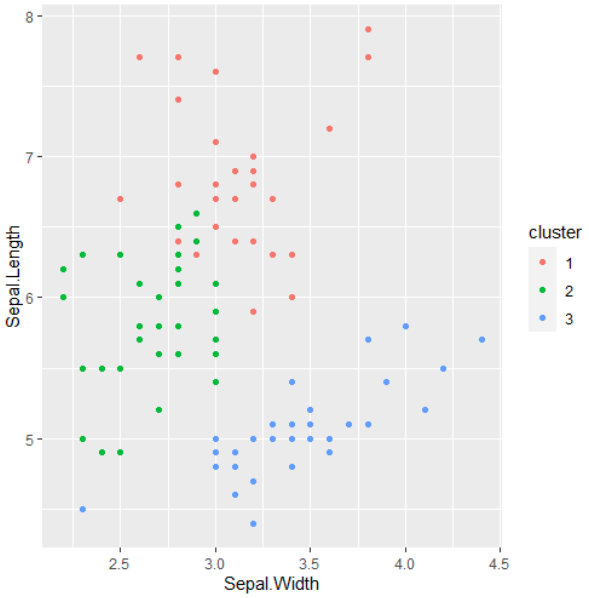
```
> iris.kmeans <- kmeans(training.data, centers = 3, iter.max = 10000)
> iris.kmeans$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1    1.1487190   0.1113613    0.9813667   0.9965297
2   -0.0443383  -0.8544827    0.3725492   0.3169439
3   -1.0021187   0.8259080   -1.3017316  -1.2552234
```

해석)

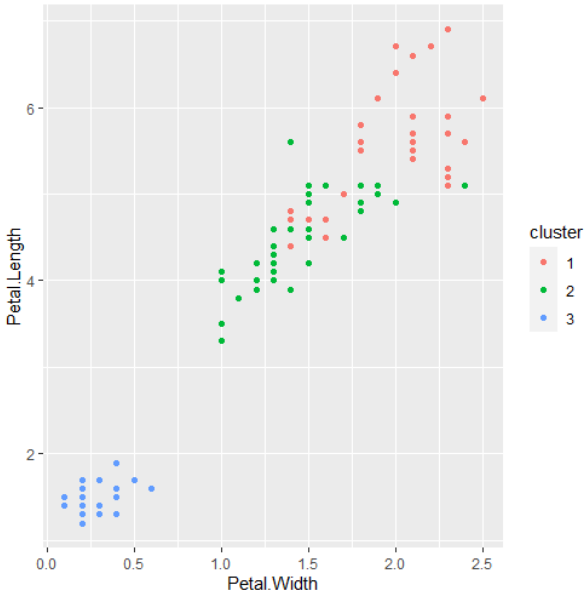
적용결과는 나왔으나 인사이트 도출을 위해서는 시각화가 필요한 모습이다.

3) 시각화

위 군집분석 결과를 시각화한 도표이다.



(그림 3-1)



(그림 3-2)

(그림 3-1)은 Sepal 을 기준으로 그린 도표이다.

(그림 3-2)은 Petal 을 기준으로 그린 도표이다.

결론

각 데이터 셋 별로 지도학습(분류/예측기법)과 비지도학습(군집분석)을 실시하였다.

지도학습 기법들은 대체적으로 준수한 정확도를 보였으나 인공신경망 기법에서는 다소 정확도의 변동폭이 컸고, 각 기법들을 여러 번 실시했을 때는 평균적으로 랜덤포레스트 기법의 정확도가 높은 편이었다.

비지도학습에서는 K-means clustering 기법을 활용하여 iris 데이터를 분류한 결과 거의 정확하게 꽃의 종류를 구분해냈다.

부록

1. 위스콘신 유방암 데이터 셋

1-1) 라이브러리 모음

```
## Project4 C3 ######################################################################
# 기간: 21.11.22(월) 9:30 까지
# C3 조: 오준서 천성한 한호종 황윤수


# 라이브러리 모음
rm(list=ls())
install.packages("caret")
install.packages("e1071")
install.packages("neuralnet")
install.packages('randomForest')
install.packages('rpart')
install.packages('neuralnet')


library(caret) # 데이터파티션()
library(dplyr) # 파이프연산자
library(e1071) # 베이즈 기법
library(randomForest) # 랜덤포레스트()
library(car)
library(rpart) # 의사결정트리
library(rpart.plot) # 트리 시각화
library(neuralnet) # 인공신경망
```

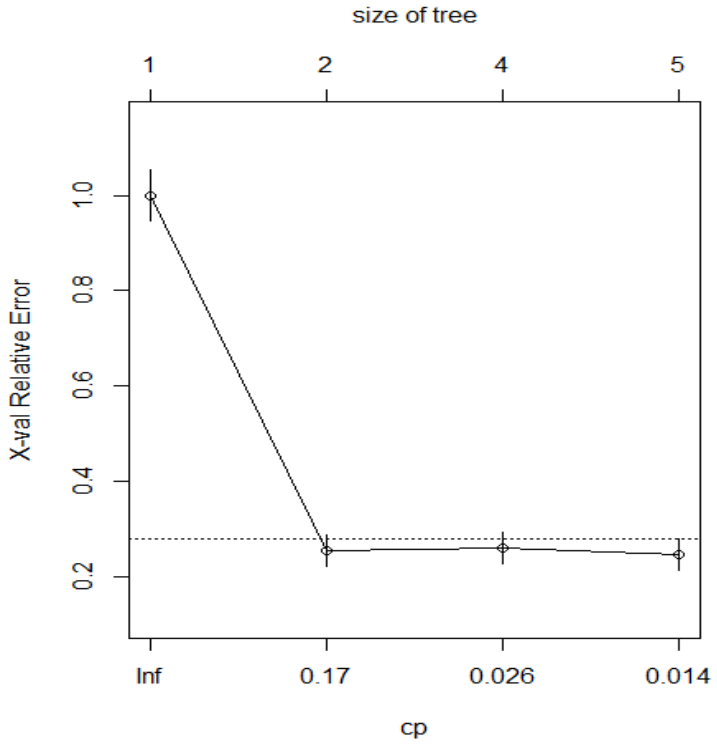
1-2) 나이브 베이즈 코드

```
# 1-2) 나이브 베이즈 ####

# (1)데이터 전처리

data1 <- read.csv('C:/rwork/wdbc.csv',header = F)

wisconsin2 <- data1 %>% select(2:12,) %>%

  `colnames<-`(c('양성여부','반경','질감','둘레','면적','매끄러움','조그만정도',

                 '오목함','오목점','대칭','프랙탈차원'))


# (2) 학습 데이터, 검정 데이터 생성

# wisconsin$양성여부 <- as.character(wisconsin$양성여부)

tr_data2 <- createDataPartition(y=wisconsin2$양성여부,p=0.7,list=FALSE)

train_Bayes <- wisconsin2[tr_data2,]

test_Bayes <- wisconsin2[-tr_data2,]


# (3) 나이브 베이즈 모델 생성

model_Bayes <- naiveBayes(양성여부~.,data=train_Bayes)

model_Bayes


# (4) 예측값 생성

predicted <- predict(model_Bayes, test_Bayes, type = "class")

table(predicted,test_Bayes$양성여부)


positiveTest <- as.factor(test_Bayes$양성여부)

confusionMatrix(predicted,positiveTest)
```

1-3) 의사결정트리 코드

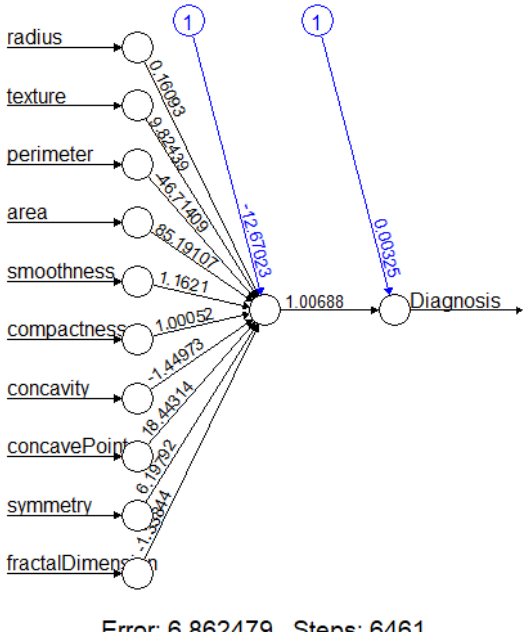
```
# 1-3)의사결정트리 ####
# (1) 데이터 전처리
data1 <- read.csv('wdbc.csv',header = F)
wisconsin3 <- data1 %>% select(1:12,) %>%
  `colnames<-`(c('id','양성여부','반경','질감','둘레','면적','매끄러움','조그만정도',
                 '오목함','오목점','대칭','프랙탈차원'))
# 자료형 변환
wisconsin3$양성여부 <- as.factor(wisconsin3$양성여부)

# (2) 분류모델 생성
model_tree <- ctree(양성여부 ~ .,data=wisconsin3)
tree <- rpart(양성여부 ~ ., data=wisconsin3)
summary(tree)

# (3) 분류분석 결과
plot(model_tree)
prp(tree, type=4, extra=2)

# (4) 교차타당성오차
tree$cptable
opt <- which.min(tree$cptable[,"xerror"])
cp <- tree$cptable[opt,"CP"]
prune.c <- prune(tree,cp=cp)
plotcp(tree)
# id 변수가 빠진 트리와의 비교
wisconsin2 <- data1 %>% select(2:12,) %>%
  `colnames<-`(c('양성여부','반경','질감','둘레','면적','매끄러움','조그만정도',
                 '오목함','오목점','대칭','프랙탈차원'))
wisconsin2$양성여부 <- as.factor(wisconsin2$양성여부)
model_tree2 <- ctree(양성여부 ~ .,data=wisconsin2)
plot(model_tree2)
```

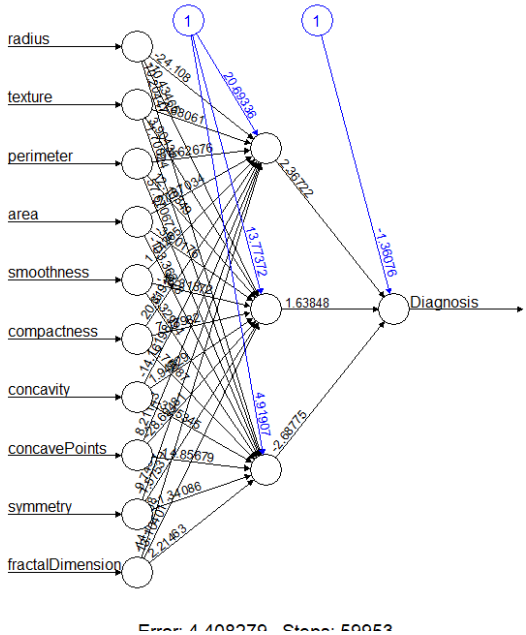
*교차 타당성 검사 시각화 그래프

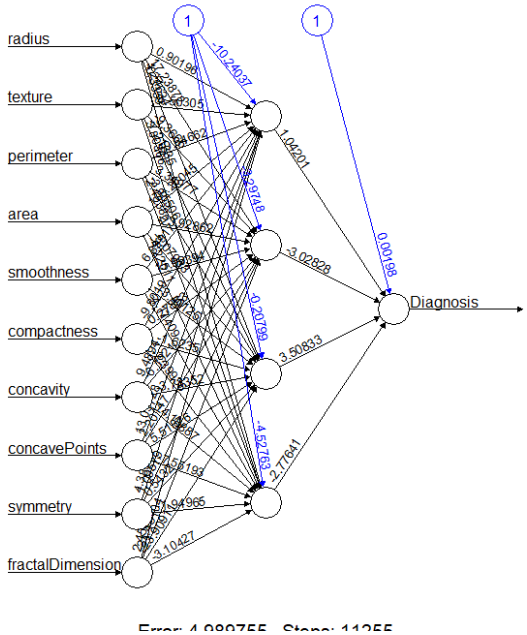


1-4) 인공신경망 코드

```
# 1-4)인공신경망 #### (원본과 다름)
# (1) 데이터셋 생성
breastCancer <- read.csv('C:/Rwork/wdbc.csv', header = F); #데이터파일
breastCancer4 <- breastCancer[c(2:12)] # 컬럼명 지정은 위와 동일
# (2) 범주형 변수를 수치형으로 변환
breastCancer4$Diagnosis[breastCancer4$Diagnosis == 'B'] <- 1
breastCancer4$Diagnosis[breastCancer4$Diagnosis == 'M'] <- 2
breastCancer4$Diagnosis <- as.integer(breastCancer4$Diagnosis)
# (3) 학습 데이터, 검정 데이터 생성
idx4 <- createDataPartition(breastCancer4$Diagnosis, p = 0.7, list = F)
training_breast4 = breastCancer4[idx4,] #위 index 값 그대로 사용
testing_breast4 = breastCancer4[-idx4,]
# (4) 데이터 정규화
normal <- function(x){
  return((x - min(x))/(max(x)-min(x)))
training_nor4 <- as.data.frame(lapply(training_breast4, normal))
testing_nor4 <- as.data.frame(lapply(testing_breast4, normal))
# (5) 인공신경망 모델 생성
formula4 = Diagnosis~.
model_net1 = neuralnet(formula4, data = training_nor4, hidden = 1)
model_net3 = neuralnet(formula4, data = training_nor4, hidden = 3)
model_net4 = neuralnet(formula4, data = training_nor4, hidden = 4)
# (6) 분류모델 성능 평가
# compute()사용
model_result1 <- compute(model_net1, testing_nor4)
model_result3 <- compute(model_net3, testing_nor4)
model_result4 <- compute(model_net4, testing_nor4)
# 상관관계분석
cor(model_result1$net.result, testing_nor4$Diagnosis)
cor(model_result3$net.result, testing_nor4$Diagnosis)
cor(model_result4$net.result, testing_nor4$Diagnosis)
```







1-4) 인공신경망 모델 시각화

1-5) 랜덤포레스트 코드

```
# 1-5)앙상블 기법-랜덤 포레스트 기법 ####
# (1) 데이터 전처리
breastCancer <- read.csv('C:/Rwork/wdbc.csv', header = F) #데이터파일
names(breastCancer) <- c('IDNumber', 'Diagnosis', 'radius', 'texture', 'perimeter',
                         'area', 'smoothness', 'compactness', 'concavity',
                         'concavePoints','symmetry', 'fractalDimension')
breastCancer.RF <- breastCancer[c(2:12)]
breastCancer.RF$Diagnosis <- as.factor(breastCancer.RF$Diagnosis) #종속변수 범주형 변환

# (2) 학습데이터, 검증데이터 생성
idx.RF <- createDataPartition(breastCancer.RF$Diagnosis, p = 0.7, list = F)
breastTrain <- breastCancer.RF[idx.RF,] #학습데이터
breastTest <- breastCancer.RF[-idx.RF,] #테스트데이터

# (3) 랜덤포레스트 모델 생성
formula1 = Diagnosis~.
rfbreast <- randomForest(formula1, data=breastTrain, ntree=100, proximity=TRUE)
#proximity=TRUE 는 개체들 간의 근접도 행렬을 제공 : 동일한 최종노드에 포함되는 빈도에 기초함

# (4) 모델 성능 확인
table(predict(rfbreast), breastTrain$Diagnosis)
plot(rfbreast)

rf.predbreast <- predict(rfbreast, newdata = breastTest)
r2 = table(rf.predbreast, breastTest$Diagnosis);r2; (r2[1,1]+r2[2,2])/nrow(breastTest)
```

2. 전복 데이터 셋

2-2) 다중회귀분석 코드

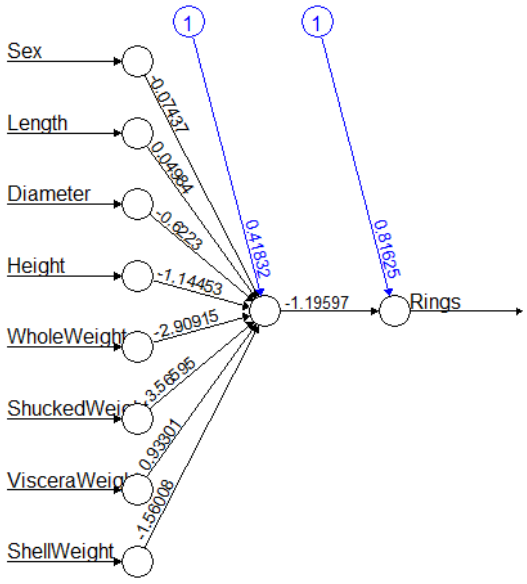
```
# 2-2)다중회귀분석 ####
# (1)데이터 전처리
abalone <- read.csv("abalone.csv", header = F)
names(abalone) <- c("Sex","Length","Diameter continuous","Height","Whole weight","Shucked weight",
                    "Viscera weight","Shell weight","Rings")
abalone2 <- abalone
abalone2$Sex[abalone2$Sex == "M"] <- 1 # 문자열 숫자로 변화
abalone2$Sex[abalone2$Sex == "F"] <- 2
abalone2$Sex[abalone2$Sex == "I"] <- 0
abalone2$Sex <- as.numeric(abalone2$Sex) # 문자열을 수열로 변환

# (2)회귀분석모델 생성
formula2 <- Rings ~ .
idx2 <- sample(1:nrow(abalone2), 0.7*nrow(abalone2))
train2 <- abalone2[idx2,] #학습데이터
test2 <- abalone2[-idx2,] #검정데이터

a_model <- lm(formula2, data = train2) # 모델생성
a_model
summary(a_model)

# (3)모델 평가
pred1 <- predict(a_model,test2) #예측치 생성
pred1

cor(pred1, test2$Rings) #회귀모델 평가
```

2-3) 의사결정트리 코드

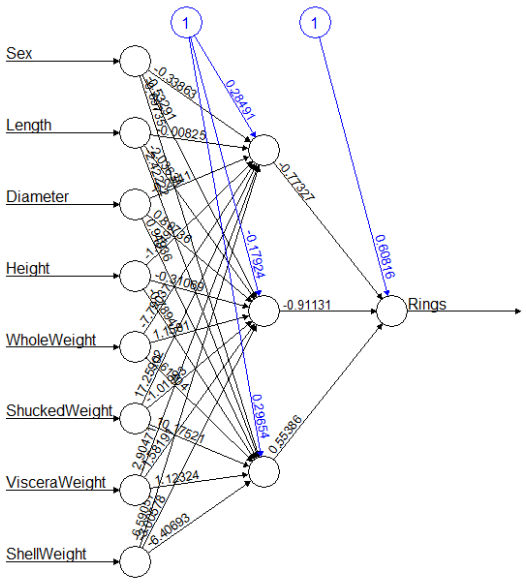
```
# 2-3)의사결정트리 ####
# (1) 데이터 준비
abalone <- read.csv('C:/Rwork/abalone.csv', header = F); #데이터파일
names(abalone) <- c('Sex', 'Length', 'Diameter', 'Height', 'WholeWeight',
                    'ShuckedWeight','VisceraWeight','ShellWeight','Rings')
abalone3 <- abalone
abalone3$Sex[abalone3$Sex == 'M'] <- 1
abalone3$Sex[abalone3$Sex == 'F'] <- 2
abalone3$Sex[abalone3$Sex == 'I'] <- 0
sexc <- abalone3$Sex
sexi <- as.integer(sexc)
abalone3$Sex <- sexi

# (2) 학습데이터, 검정데이터 생성
idx3 <- createDataPartition(abalone3$Rings, p = 0.7, list = F)
abaloneTrain3 <- abalone3[idx3,] #학습데이터
abaloneTest3 <- abalone3[-idx3,] #테스트데이터

# (3) 의사결정트리(Decision Tree) 모델 생성
rpartFit <- rpart(Rings ~ ., data = abaloneTrain3)
summary(rpartFit)

# (4) 시각화 및 성능평가
rpart.plot(rpartFit, digits = 3, type = 0, extra = 1, fallen.leave = F, cex = 1)

rpartPre <- predict(rpartFit, newdata = abaloneTest3)
```

2-4) 인공신경망 코드

```
# 2-4)인공신경망 #### (원본과 다름)
# (1)데이터 전처리
Abalone4 <- read.csv('C:/Rwork/abalone.csv', header = F); #데이터파일 # 컬럼명 지정은 위와 동일
abalone4$Sex[abalone4$Sex == 'M'] <- 1
abalone4$Sex[abalone4$Sex == 'F'] <- 2
abalone4$Sex[abalone4$Sex == 'I'] <- 0
sexc <- abalone4$Sex
sexi <- as.integer(sexc)
abalone4$Sex <- sexi
idx4 <- createDataPartition(abalone4$Rings, p = 0.7, list = F)
abaloneTrain <- abalone4[idx4,] #학습데이터
abaloneTest <- abalone4[-idx4,] #테스트데이터
# (2)데이터 정규화
normal <- function(x){
  return((x - min(x))/(max(x)-min(x)))}
# (3)학습데이터, 검정데이터 생성
training_nor4 <- as.data.frame(lapply(abaloneTrain, normal))
testing_nor4 <- as.data.frame(lapply(abaloneTest, normal))
# (4)인공신경망 모델 생성
model_net41 = neuralnet(Rings ~ ., data = training_nor4, hidden = 1)
model_net43 = neuralnet(Rings ~ ., data = training_nor4, hidden = 3)
# (5)시각화
plot(model_net41)
plot(model_net43)
# (6)분류모델 성능 평가
# compute()사용
model_result41 <- compute(model_net41, testing_nor4[-9])
model_result43 <- compute(model_net43, testing_nor4[-9])
# 상관관계를 통한 정확도 평가
cor(model_result41$net.result, testing_nor4$Rings)
cor(model_result43$net.result, testing_nor4$Rings)
```



Error: 8.94278 Steps: 9301



Error: 7.842203 Steps: 36285

*은닉층이 2 개일 때와 3 개일 때의 Steps 의 차이 및 가중치 참고

2-5) 랜덤포레스트 코드

```
# 2-5)앙상블 기법-랜덤 포레스트 기법 ####
# (1) 데이터 전처리
abalone5 <- read.csv('C:/Rwork/abalone.csv', header = F); #데이터파일
names(abalone5) <- c('Sex', 'Length', 'Diameter', 'Height', 'WholeWeight',
                     'ShuckedWeight','VisceraWeight','ShellWeight','Rings')

# (2) 학습데이터, 테스트데이터 생성
idx5 <- createDataPartition(abalone5$Rings, p = 0.7, list = F)
abaloneTrain <- abalone5[idx5,] #학습데이터
abaloneTest <- abalone5[-idx5,] #테스트데이터

# (3) 랜덤포레스트 모델 생성
rfAba <- randomForest(Rings ~ ., data=abaloneTrain, ntree=100, proximity=TRUE) #컬럼명에 띄어쓰기 있으면 오류
#proximity=TRUE 는 개체들 간의 근접도 행렬을 제공 : 동일한 최종노드에 포함되는 빈도에 기초함
table(predict(rfAba), abaloneTrain$Rings)
rfAba

# (4) 중요 변수 출력 및 시각화
importance(rfAba)

varImpPlot(rfAba)
plot(rfAba)

# (5) 모델 평가
rf.predAba <- predict(rfAba, newdata = abaloneTest)
table(rf.predAba, abaloneTest$Rings)
cor(rf.predAba, abaloneTest$Rings)
```

참조

1. Wisconsin Breast Cancer Data Sets

data.csv # 데이터 셋 header = TRUE

wdbc.csv # 데이터 셋 header = FALSE

wdbc.readme.txt # 데이터 셋 설명

no1_나이브베이즈.R

no1_랜덤포레스트.R

no1_의사결정트리.R

no1_인공신경망.R

2. abalone Data Sets

ablone # 데이터 셋

ablone.readme.txt #데이터 셋 설명

no2_다중회귀분석.R

no2_랜덤포레스트.R

no2_의사결정트리.R

no2_인공신경망.R

3. iris data sets

no3_k-means_clustering

https://blog.daum.net/sys4ppl/7 # 군집분석 참고자료

종합코드

Project4_C3_오준서,천성한,한호종,황윤수.R