01. IRIS 데이터 셋을 이용한 머신러닝

In [6]:
```
library(caret)
```

In [7]:
```
idx<-createDataPartition(iris$Species, p=0.7, list=F)
```

In [8]:
```
iris_train<-iris[idx, ] #생성된 인덱스를 이용, 70%의 비율로 학습용 데이터 세트 추출
iris_test<-iris[-idx, ] #생성된 인덱스를 이용, 30%의 비율로 평가용 데이터 세트 추출
table(iris_train$Species) #학습용 데이터의 목표변수(Species)의 빈도 분포
```

```
    setosa versicolor  virginica 
        35         35         35 
```

In [9]:
```
table(iris_test$Species) #평가용 데이터의 목표변수(Species)의 빈도 분포
```

```
    setosa versicolor  virginica 
        15         15         15 
```

02. 나이브 베이즈 모형 사용해 보기

- e1071 패키지의 naiveBayes 함수
- klaR 패키지의 NaiveBayes 함수

사용 예

모델 생성

model<-naiveBayes(iris_train, class = iris_train$Species, laplace=1)
* train인자에는 훈련 데이터 세트를 입력
* class는 범주형 목표변수를 입력
* laplace인자는 나이브 베이즈 알고리즘 적용 시 특정 속성범주의 발생확률이 0이 될 경우 전체 추정결과가 왜곡되는 것을 방지하기 위해 작은 값을 추가하는 것으로 일종의 보정 인자

결과 예측

result<-predict(model, iris_test, type="class")
* model은 (나)의 훈련결과를 통해 도출된 모델객체 명
* type 인자는 예측된 결과의 출력형태를 의미
  type 인자 값이 "class"일 경우 예측된 범주 값이 도출되고 "raw"일 경우 예측 확률값이 도출

In [21]:
```
library(e1071) #나이브 베이즈 기법 적용하기 위한 e1071 패키지 로드
```

In [22]:
```
naive.result<-naiveBayes(iris_train, iris_train$Species,laplace=1) #나이브 베이즈 적합
```

In [23]:
```
naive.pred<-predict(naive.result, iris_test, type="class") #테스트 데이터 평가
```

In [24]:
```
table(naive.pred, iris_test$Species) #분류 결과 도출
```

```
naive.pred    setosa versicolor virginica
  setosa          15          0         0
  versicolor       0         15         0
  virginica        0          0        15
```

분류 결과 정확도 1의 상당한 정확도를 자랑한다.

In [25]:
```
confusionMatrix(naive.pred, iris_test$Species)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         15         0
  virginica       0          0        15

Overall Statistics

               Accuracy : 1
                 95% CI : (0.9213, 1)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            1.0000           1.0000
Specificity                 1.0000            1.0000           1.0000
Pos Pred Value              1.0000            1.0000           1.0000
Neg Pred Value              1.0000            1.0000           1.0000
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3333           0.3333
Detection Prevalence        0.3333            0.3333           0.3333
Balanced Accuracy           1.0000            1.0000           1.0000
```

03. 로지스틱 회귀분석 기법 사용해 보기

- 현재 실습에 사용하고 있는 iris 데이터 세트의 목표변수 Y값의 범주는 3개여서 다항 로지스틱 회귀가 되므로 이 경우에는 nnet 패키지에서 제공하는 multinom 함수를 사용하여 로지스틱 회귀
- glm은 목표변수 Y값이 이항 형태인 이항 로지스틱 회귀분석에만 사용 가능

사용 예시

모델 만들기

model<-multinom(Species ~ . data=iris_train)
* formula 인자에 Species ~ Sepal.Length+Sepal.Width+Petal.Length+Petal.Width로 입력
* 모든 설명변수를 formula에 투입할 경우, 전체 변수명을 나열하기 보다는 마침표 '.' 인자를 사용

예측

result<-predict(model, iris_test)
* model은 (나)의 훈련결과를 통해 도출된 모델객체명

In [26]:
```
library(nnet) #다항 로지스틱 회귀를 사용하기 위한 nnet 패키지 로딩
```

In [27]:
```
multi.result<-multinom(Species~., iris_train) #훈련 데이터 통한 모형 적합
```

```
# weights:  18 (10 variable)
initial  value 115.354290
iter  10 value 12.003392
iter  20 value 5.423791
iter  30 value 5.016135
iter  40 value 4.954630
iter  50 value 4.949929
iter  60 value 4.943645
iter  70 value 4.941831
iter  80 value 4.941317
iter  90 value 4.941261
iter 100 value 4.941242
final  value 4.941242
stopped after 100 iterations
```

In [28]:
```
multi.pred<-predict(multi.result, iris_test) #테스트 데이터 이용한 평가
```

In [29]:
```
table(multi.pred, iris_test$Species) #분류 결과도출
```

```
multi.pred   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         15         1
  virginica       0          0        14
```

In [30]:
```
confusionMatrix(multi.pred, iris_test$Species)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         15         1
  virginica       0          0        14

Overall Statistics

               Accuracy : 0.9778
                 95% CI : (0.8823, 0.9994)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9667
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            1.0000           0.9333
Specificity                 1.0000            0.9667           1.0000
Pos Pred Value              1.0000            0.9375           1.0000
Neg Pred Value              1.0000            1.0000           0.9677
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3333           0.3111
Detection Prevalence        0.3333            0.3556           0.3111
Balanced Accuracy           1.0000            0.9833           0.9667
```

04. 의사결정트리 기법 사용

- rpart 패키지(rpart 함수), party 패키지(ctree 함수), C50 패키지(C5.0 함수)
- rpart는 대표적인 의사결정트리 기법인 CART(Classification and Regression Trees)를 구현한 패키지
- party 패키지의 ctree 함수는 CART 알고리즘의 문제점을 해결하기 위해 조건부 추론 트리를 구현한 함수
- C50 패키지의 C5.0 함수는 엔트로피 지수를 사용하는 C5.0 알고리즘을 구현한 함수

In [31]:
```
library(rpart) #의사결정트리 기법을 사용하기 위한 rpart 패키지 로딩
```

In [32]:
```
rpart.result<-rpart(Species~., iris_train) #훈련데이터 통한 모형 적합
```

In [33]:
```
rpart.result
```

```
n= 105

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

1) root 105 70 setosa (0.3333333 0.3333333 0.3333333)
  2) Petal.Length< 2.35 35  0 setosa (1.0000000 0.0000000 0.0000000) *
  3) Petal.Length>=2.35 70 35 versicolor (0.0000000 0.5000000 0.5000000)
    6) Petal.Width< 1.75 38  4 versicolor (0.0000000 0.8947368 0.1052632) *
    7) Petal.Width>=1.75 32  1 virginica (0.0000000 0.0312500 0.9687500) *
```

In [35]:
```
rpart.pred<-predict(rpart.result, iris_test, type="class") #테스트 데이터 이용 평가
```

In [36]:
```
table(rpart.pred, iris_test$Species) #분류 결과도출
```

```
rpart.pred   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         15         1
  virginica       0          0        14
```

In [37]:
```
confusionMatrix(rpart.pred, iris_test$Species)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         15         1
  virginica       0          0        14

Overall Statistics

               Accuracy : 0.9778
                 95% CI : (0.8823, 0.9994)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9667
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            1.0000           0.9333
Specificity                 1.0000            0.9667           1.0000
Pos Pred Value              1.0000            0.9375           1.0000
Neg Pred Value              1.0000            1.0000           0.9677
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.3333           0.3111
Detection Prevalence        0.3333            0.3556           0.3111
Balanced Accuracy           1.0000            0.9833           0.9667
```

05. 서포트 벡터 머신 기법 사용

- e1071 패키지의 svm 함수
- klaR 패키지의 svmlight 함수
- kernlab 패키지의 ksvm 함수

사용 예시

모델 만들기

model<-ksvm(Species ~ . , data= iris_train, kernel= “rbfdot")
* formula 인자에 Species ~ . 형태로 목표변수와 설명변수들을 입력
* kernel 인자는 커널함수를 지정하는 인자로서 “rbfdot” (가우시안 RBF 커널)
* "polydot" (polynomial 커널),
* "tanhdot"(하이퍼볼릭 탄젠트 시그모이드 커널),
* "vanilla dot"(linear 커널 : 특별한 변환 없이 내적을 계산) 등이 있으며
* 기본값은 "rbfdot"", 즉 가우시안 RBF 커널

예측

result<-predict(model, iris_test, type= “response”)
* model은 훈련결과를 통해 도출된 모델객체명
* type 인자에는 "response"(예측된 범주 분류 값)와 "probabilities"(예측된 확률값)

In [39]:
```
library(kernlab) #서포트 벡터 머신 기법을 사용하기 위한 kernlab 패키지 로딩
```

```
Attaching package: 'kernlab'

The following object is masked from 'package:ggplot2':

    alpha
```

In [40]:
```
svm.result<-ksvm(Species~., iris_train, kernel="rbfdot") #훈련 데이터 통한 모형적합
```

In [41]:
```
svm.pred<-predict(svm.result, iris_test, type="response") #테스트 데이터 평가
```

In [42]:
```
table(svm.pred, iris_test$Species) #분류 결과도출
```

```
svm.pred     setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         13         0
  virginica       0          2        15
```

In [43]:
```
confusionMatrix(svm.pred, iris_test$Species)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         13         0
  virginica       0          2        15

Overall Statistics

               Accuracy : 0.9556
                 95% CI : (0.8485, 0.9946)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9333
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.8667           1.0000
Specificity                 1.0000            1.0000           0.9333
Pos Pred Value              1.0000            1.0000           0.8824
Neg Pred Value              1.0000            0.9375           1.0000
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.2889           0.3333
Detection Prevalence        0.3333            0.2889           0.3778
Balanced Accuracy           1.0000            0.9333           0.9667
```

06. 서포트 벡터 머신 기법 사용

- randomForest 패키지의 randomForest 함수

사용 예시

모델 만들기

```
randomForest(formula, data, ntree = 500, mtry, importance)
  * formula 인자에는 목표변수 ~ 독립변수들 형태로 입력
  * ntree 인자는 랜덤하게 사용할 의사결정트리의 개수 (기본 500개)
  * mtry는 노드를 나눌 때 무작위로 선택할 입력변수의 개수를 명시
  * importance는 랜덤 포레스트 추정결과 각 변수의 중요도를 평가
```

예측

```
model<-randomForest(Species ~ . , data= iris_train, ntree=500)
  * formula 인자에 Species ~ . 형태로 목표변수와 설명변수들을 입력
  * 일반적으로는 예측하고자 하는 값이 복잡하고 데이터양과 변수가 많을수록 더 큰 트리 개수를 사용

result<-predict(model, iris_test, type="response")
  * model은 훈련된 모델객체 명
  * type 인자에는 "response"(예측된 범주 분류)와 "prob"(예측된 확률값), "votes"(투표결과 행렬)가 있는데 일반적으로 respons
e를 많이 사용
```

In [45]:
```
library(randomForest) #랜덤 포레스트 기법을 사용하기 위한 kernlab 패키지 로딩
```

```
randomForest 4.6-12
Type rfNews() to see new features/changes/bug fixes.

Attaching package: 'randomForest'

The following object is masked from 'package:ggplot2':

    margin
```

In [46]:
```
rf.result<-randomForest(Species~., iris_train, ntree=500) #훈련 데이터 통한 모형적합
```

In [47]:
```
rf.pred<-predict(svm.result, iris_test, type="response") #테스트 데이터 이용 평가
```

In [48]:
```
table(rf.pred, iris_test$Species) #분류 결과도출
```

```
rf.pred      setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         13         0
  virginica       0          2        15
```

In [49]:
```
confusionMatrix(svm.pred, iris_test$Species)
```

```
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         13         0
  virginica       0          2        15

Overall Statistics

               Accuracy : 0.9556
                 95% CI : (0.8485, 0.9946)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9333
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: setosa Class: versicolor Class: virginica
Sensitivity                 1.0000            0.8667           1.0000
Specificity                 1.0000            1.0000           0.9333
Pos Pred Value              1.0000            1.0000           0.8824
Neg Pred Value              1.0000            0.9375           1.0000
Prevalence                  0.3333            0.3333           0.3333
Detection Rate              0.3333            0.2889           0.3333
Detection Prevalence        0.3333            0.2889           0.3778
Balanced Accuracy           1.0000            0.9333           0.9667
```

Summary

(가) 각 기법 내에서도 다양한 파라미터 조정과 세부 설정 등을 통해 다른 결과가 도출될 수 있음

(나) 의사결정트리, 나이브 베이즈, 로지스틱 회귀가 높게 나옴.

(다) 모형 분류결과를 평가하는 지표에 관해서도 모형 평가 지표가 정확도(accuracy)만 있는 것은 아니며,

민감도, 정밀도 등 다양한 평가 지표가 존재하므로, 정확도만으로 어떤 특정 기법이 더 우월하다고 평가하는 것은 곤란

In [ ]: