Wk12-3 : 랜덤포레스트(Random Forest)

1. 랜덤포레스트 (Random Forest)-모형설명

- 랜덤포레스트 (Random Forest)

  - 2001년에 Leo Breiman에 의해 제안된 기법 의사결정나무의 단점(과적합)을 개선한 알고리즘

  - Ensemble 기법을 사용한 모델로서 주어진 데이터로 리샘플링을 통해 다수의 의사결정나무를 만든 다음, 여러 모델의 예측 결과들을 종합해 정확도를 높이는 방법

training data로부터 표본의 크기가 n인 bootstrap sample을 추출

↓

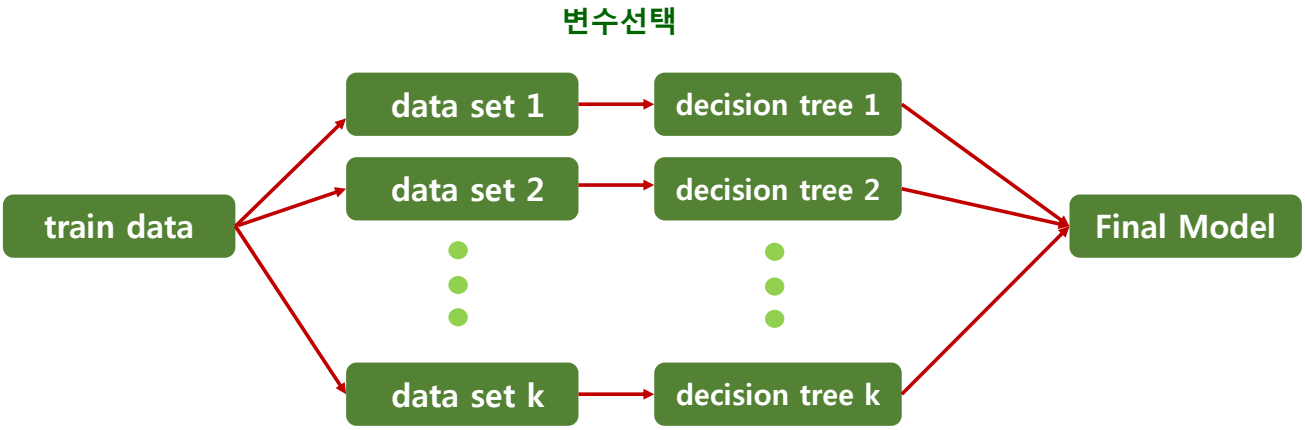
tree모형 구성 (tree1, tree2, ...treek)

↓

각 모델 tree들의 앙상블 결과를 출력

1. 랜덤포레스트 (Random Forest) - 모형설명

- Bagging(Bootstrap Aggregating)
  - 전체 데이터에서 학습데이터를 복원추출(resampling) 트리를 구성
  - Training Data에서 Random Sampling



2. 랜덤포레스트 (Random Forest)

- 랜덤포레스트 (Random forest) 패키지 : randomForest

```
# lec12_3_rf.R
# Random Forest using R

# random forest package
install.packages("randomForest")
library(randomForest)
help(randomForest)

# load caret package for confusion matrix
library(caret)
```

randomForest 패키지 설치, 라이브러리 설정

caret 라이브러리 설정 (ConfusionMatrix)

2. 랜덤포레스트 (Random Forest)

- **help(randomForest)**

ntree: Number of decision trees to be grown
replace: Takes True and False and indicates whether to take sample with/without replacement
sampsize: Sample size to be drawn from the input data for growing decision tree
importance: Whether independent variable importance in random forest be assessed

Files | Plots | Packages | Help | Viewer

R: Classification and Regression with Random Forest Find in Topic

randomForest {randomForest} R Documentation

Classification and Regression with Random Forest

Description

`randomForest` implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression. It can also be used in unsupervised mode for assessing proximities among data points.

Usage

```
## S3 method for class 'formula'
randomForest(formula, data=NULL, ..., subset, na.action=na.fail)
## Default S3 method:
randomForest(x, y=NULL,  xtest=NULL, ytest=NULL, ntree=500,
             mtry=if (!is.null(y) && !is.factor(y))
             max(floor(ncol(x)/3), 1) else floor(sqrt(ncol(x))),
             replace=TRUE, classwt=NULL, cutoff, strata,
             sampsize = if (replace) nrow(x) else ceiling(.632*nrow(x)),
             nodesize = if (!is.null(y) && !is.factor(y)) 5 else 1,
             maxnodes = NULL,
             importance=FALSE, localImp=FALSE, nPerm=1,
             proximity, oob.prox=proximity,
             norm.votes=TRUE, do.trace=FALSE,
             keep.forest=!is.null(y) && is.null(xtest), corr.bias=FALSE,
             keep.inbag=FALSE, ...)
## S3 method for class 'randomForest'
print(x, ...)
```


2. 랜덤포레스트 (Random Forest)

- **iris 데이터 (iris.csv)**

input변수(독립변수) **output변수(종속변수, 타겟변수)**

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3	1.4	0.1	setosa

타겟변수(y) : setosa, versicolor, virginica

Iris setosa Iris versicolor Iris virginica

2. 랜덤포레스트 (Random Forest)

- iris 데이터 (학습데이터와 검증데이터의 분할)

```
# set working directory
setwd("D:/tempstore/moocr/wk12")

# read csv file
iris<-read.csv("iris.csv")
attach(iris)

# training/ test data : n = 150
set.seed(1000)
N<-nrow(iris)
tr.idx<-sample(1:N, size=N*2/3, replace=FALSE)

# split training and test data
train<-iris[tr.idx,]
test<-iris[-tr.idx,]
#dim(train)
#dim(test)
```

데이터분할 (학습데이터 2/3, 검증데이터 1/3)

train (100개의 데이터)
test (50개의 데이터)

2. 랜덤포레스트 (Random Forest)

- 랜덤포레스트 : randomForest(종속변수~x1+x2+x3+x4, data=)

```
# Random Forest : mtry=2 (default=sqrt(p))
rf_out1<-randomForest(Species~.,data=train, importance=T)
rf_out1
```

mtry=number of variables randomly sampled as candidates at each split, default=sqrt(p)

```
> rf_out1<-randomForest(Species~.,data=train, importance=T)
> rf_out1

Call:
 randomForest(formula = Species ~ ., data = train, importance = T)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 5%
Confusion matrix:
            setosa versicolor virginica class.error
setosa          31          0         0  0.00000000
versicolor       0         29         2  0.06451613
virginica        0          3        35  0.07894737
```

2. 랜덤포레스트 (Random Forest)

- 랜덤포레스트 : randomForest(종속변수~x1+x2+x3+x4, data=)

```
# Random Forest : mtry=4
rf_out2<-randomForest(Species~.,data=train, importance=T,mtry=4)
rf_out2
```

mtry=number of variables randomly sampled as candidates at each split, default=sqrt(p)

```
> rf_out2<-randomForest(Species~.,data=train, importance=T,mtry=4)
> rf_out2

Call:
 randomForest(formula = Species ~ ., data = train, importance = T,
y = 4)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 4%
Confusion matrix:
            setosa versicolor virginica class.error
setosa          31          0         0  0.00000000
versicolor       0         30         1  0.03225806
virginica        0          3        35  0.07894737
```

mtry=4일때가
mty=2일때보다 정확도 높음

2. 랜덤포레스트 (Random Forest)

- 변수의 중요도 : random forest결과로부터 중요변수 확인

```
# important variables for RF
round(importance(rf_out2), 2)
```

```
> round(importance(rf_out2), 2)
             setosa versicolor virginica MeanDecreaseAccuracy
Sepal.Length   0.00       2.57      0.55                 2.17
Sepal.Width    0.00      -0.77     -4.53                -4.35
Petal.Length  23.60      35.35     18.87                32.31
Petal.Width   28.07      38.96     27.96                38.41
             MeanDecreaseGini
Sepal.Length             0.58
Sepal.Width              0.28
Petal.Length            27.46
Petal.Width             37.39
```

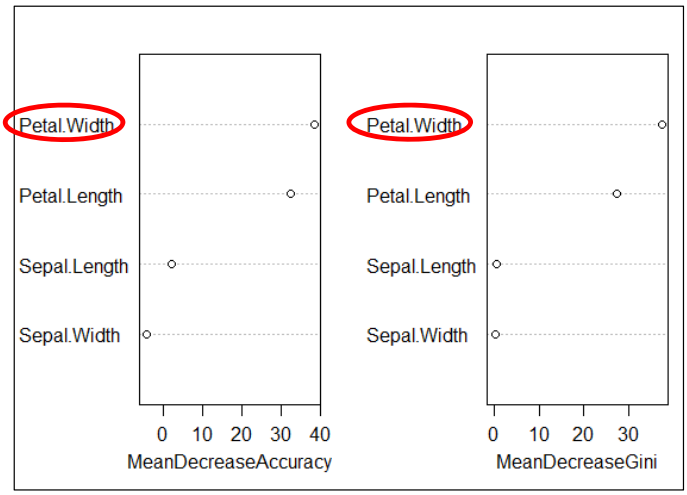
분류의 정확도에 기여도가 높은 변수

2. 랜덤포레스트 (Random Forest)

- 변수의 중요도 : random forest결과로부터 중요변수 확인

```
randomForest::importance(rf_out2)
varImpPlot(rf_out2)
```

```
> randomForest::importance(rf_out2)
               setosa versicolor virginica MeanDecreaseAccuracy
Sepal.Length  0.00000  2.5749313  0.545509             2.170898
Sepal.Width   0.00000 -0.7723049 -4.525001            -4.345987
Petal.Length 23.60032 35.3502332 18.865269            32.312349
Petal.Width  28.07243 38.9584172 27.961399            38.406958
             MeanDecreaseGini
Sepal.Length        0.5823952
Sepal.Width         0.2834697
Petal.Length       27.4559829
Petal.Width        37.3934722
```



2. 랜덤포레스트 (Random Forest)

- 랜덤포레스트 결과 정확도 : test data에 대한 정확도

```
#measuring accuracy(rf)
rfpred<-predict(rf_out2, test)
confusionMatrix(rfpred,test$Species)
```

```
> rfpred<-predict(rf_out2, test)
> confusionMatrix(rfpred,test$Species)
Confusion Matrix and Statistics

            Reference
Prediction   setosa versicolor virginica
  setosa         19          0         0
  versicolor      0         17         1
  virginica       0          2        11

Overall Statistics

               Accuracy : 0.94
```

정확도 94%