INFS4203/7203 Data Mining
The University of Queensland, Australia
Semester 2, 2018

Tutorial Week 10: Classification with R

Chandra Prasetyo Utomo

c.utomo@uq.edu.au

Objectives

1. To be able to divide dataset into training and test data.
2. To be able to implement decision tree algorithm.
3. To be able to implement K-NN algorithm.
4. To be able to evaluate classification results.

Outline

1. Experimental Data Creation (*10 minutes*)
2. Decision Tree Algorithm Implementation (*15 minutes*)
3. K-NN Algorithm Implementation (*15 minutes*)
4. Evaluation Computation (*10 minutes*)

Part 1

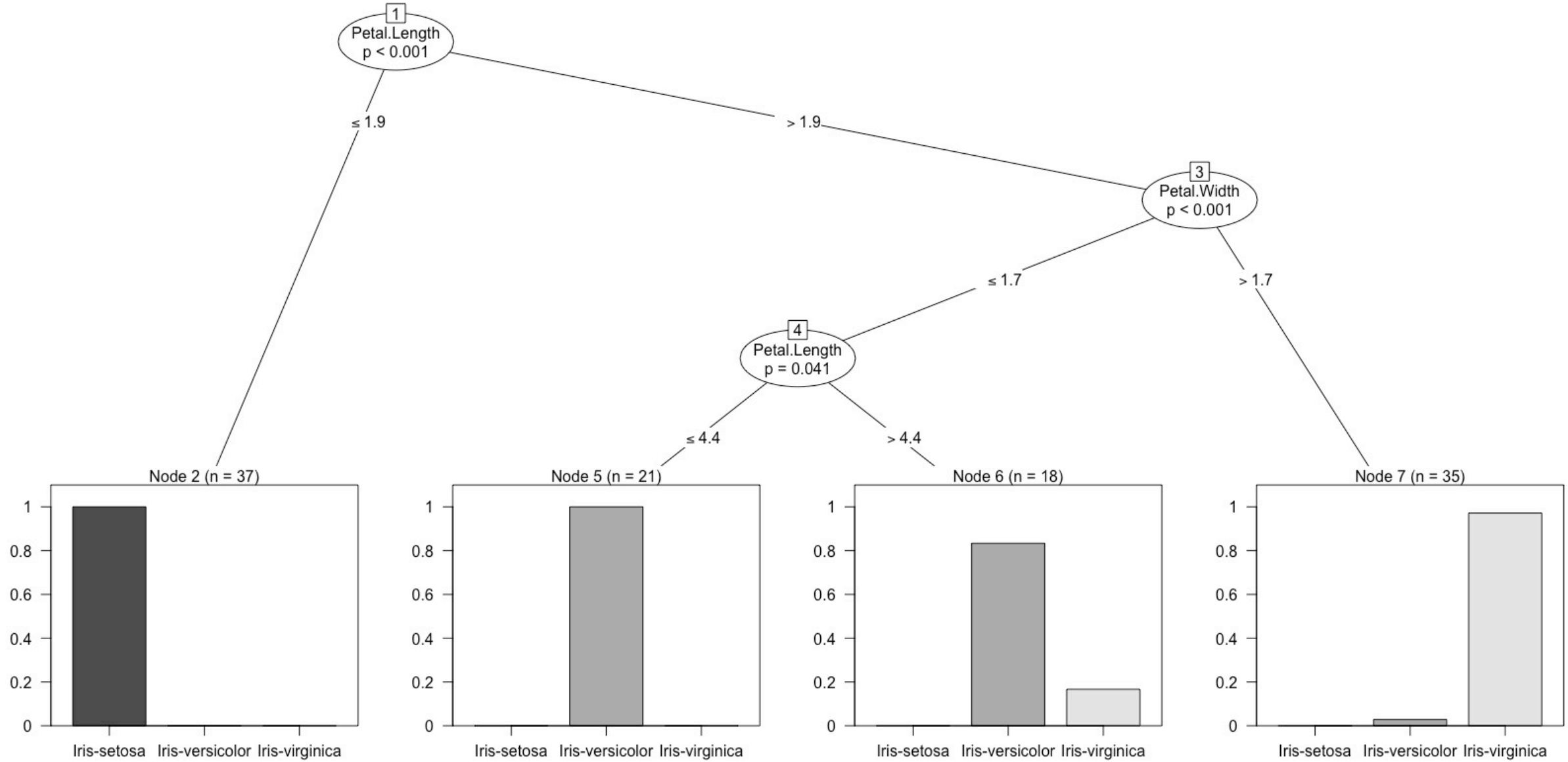
Experimental Data

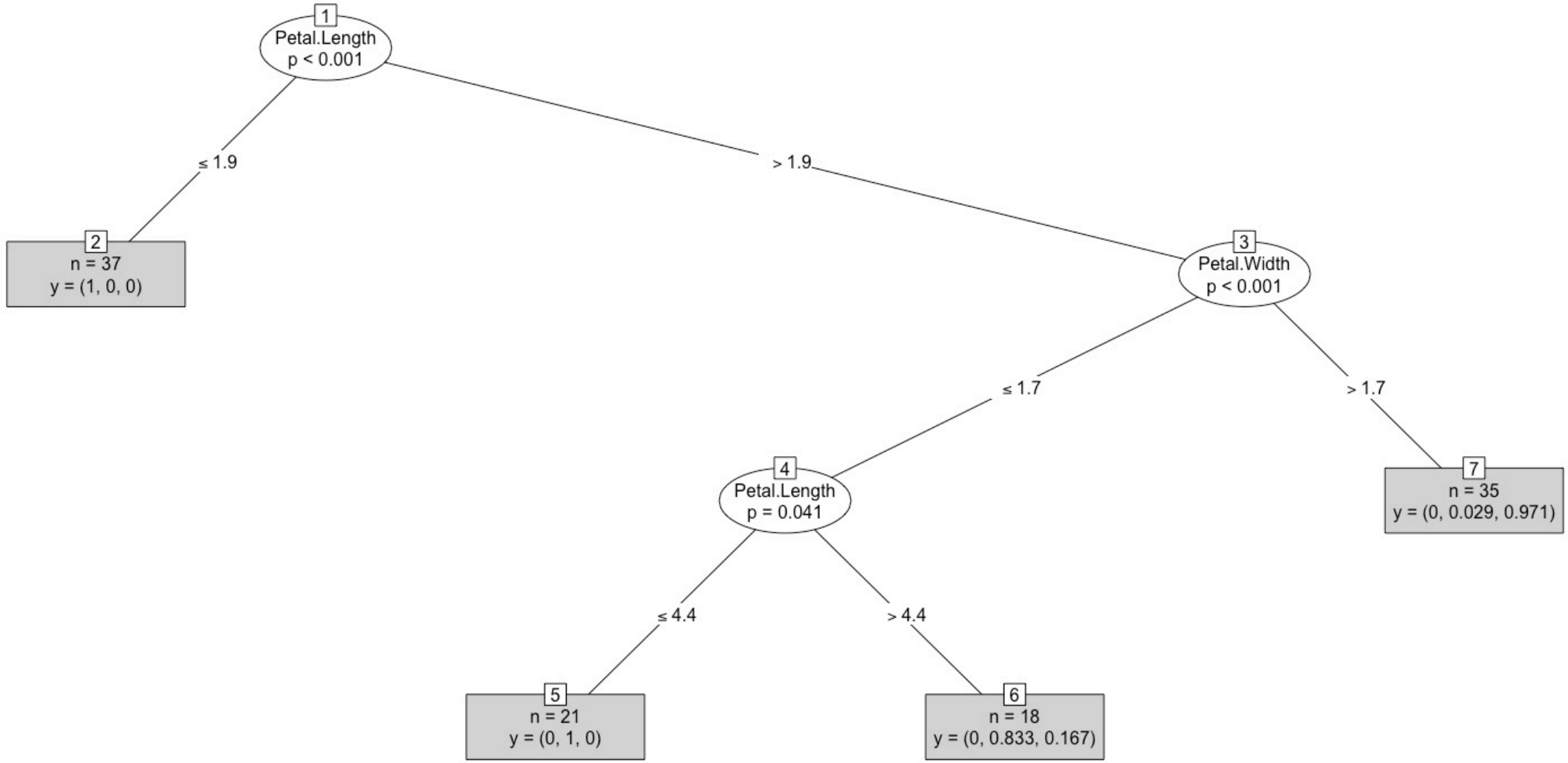
```r
## Practical Tutorial 4: Classification ##
## Part 4A: Create Experimental Data ##

# Extract Data
iris <- read.table("./data/iris.data", sep = ',')

# Assign name to variables
names(iris) <- c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width", "Species")

# For reproducible result
set.seed(2018)

# Set training and test ratio
m = nrow(iris)
training_percentage = 0.7
test_percentage = 0.3

# Sample random index
ind <- sample(2, m, replace = TRUE, prob = c(training_percentage, test_percentage))

# Select training and test data
training_data = iris[ind == 1, ]
test_data = iris[ind == 2, ]

# Save datasets to files
saveRDS(training_data, file="./data/training_data.Rda")
saveRDS(test_data, file="./data/test_data.Rda")
```

Part 2

Decision Tree Algorithm

```r
## Practical Tutorial 4: Classification ##
## Part 4B: Decision Tree ##

# load training and test data
training_data <- readRDS(file="./data/training_data.Rda")
test_data <- readRDS(file="./data/test_data.Rda")

# divide features and labels
training_features <- training_data[,1:4]
training_labels <- training_data[,5]
test_features <- test_data[,1:4]
test_labels <- test_data[,5]

# install and import "party" library
install.packages("party")
library(party)

# specify target (class) and predictors (features)
myFormula <- Species ~ Sepal.Length + Sepal.Width +
  Petal.Length + Petal.Width

# generate classification tree
iris_ctree <- ctree(myFormula, data = training_data)

# visualise the tree
plot(iris_ctree)
plot(iris_ctree, type="simple")

# predict test labels
ctree_pred <- predict(iris_ctree, newdata = test_features)
saveRDS(ctree_pred, file="./data/ctree_pred.Rda")
```


1
Petal.Length
p < 0.001
≤ 1.9
> 1.9
3
Petal.Width
p < 0.001
≤ 1.7
> 1.7
4
Petal.Length
p = 0.041
≤ 4.4
> 4.4
Node 2 (n = 37)
Node 5 (n = 21)
Node 6 (n = 18)
Node 7 (n = 35)
1
0.8
0.6
0.4
0.2
0
Iris-setosa
Iris-versicolor
Iris-virginica


1
Petal.Length
p < 0.001
≤ 1.9
> 1.9
2
n = 37
y = (1, 0, 0)
3
Petal.Width
p < 0.001
≤ 1.7
> 1.7
4
Petal.Length
p = 0.041
7
n = 35
y = (0, 0.029, 0.971)
≤ 4.4
> 4.4
5
n = 21
y = (0, 1, 0)
6
n = 18
y = (0, 0.833, 0.167)

Part 3

K-NN Algorithm

```r
## Practical Tutorial 4: Classification ##
## Part 4C: K-NN ##

# load training and test data
training_data <- readRDS(file="./data/training_data.Rda")
test_data <- readRDS(file="./data/test_data.Rda")

# divide features and labels
training_features <- training_data[,1:4]
training_labels <- training_data[,5]
test_features <- test_data[,1:4]
test_labels <- test_data[,5]

# install and import "class" library
install.packages("class")
library(class)

# classify using K-NN
knn_pred <- knn(train = training_features,
                test = test_features,
                cl = training_labels,
                k = 3)

# save actual and predicted labels
saveRDS(test_labels, file="./data/test_labels.Rda")
saveRDS(knn_pred, file="./data/knn_pred.Rda")
```

Part 4

Evaluation Computation

```r
## Practical Tutorial 4: Classification ##
## Part 4D: Evaluation ##

# load actual and predicted labels
test_labels <- readRDS(file="./data/test_labels.Rda")
ctree_pred <- readRDS(file="./data/ctree_pred.Rda")
knn_pred <- readRDS(file="./data/knn_pred.Rda")

# create the confusion matrix
cm = as.matrix(table(Actual = test_labels, Predicted = ctree_pred))

n = sum(cm) # number of instances
nc = nrow(cm) # number of classes
diag = diag(cm) # number of correctly classified instances per class
rowsums = apply(cm, 1, sum) # number of instances per class
colsums = apply(cm, 2, sum) # number of predictions per class

# compute accuracy, precision, recall, and f1
accuracy = sum(diag) / n
precision = diag / colsums
recall = diag / rowsums
f1 = 2 * precision * recall / (precision + recall)

results <- data.frame(precision, recall, f1)
accuracy
results
```

```
> cm
                 Predicted
Actual            Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              13               0              0
  Iris-versicolor           0              13              0
  Iris-virginica            0               2             11
> accuracy
[1] 0.9487179
> results
                precision    recall        f1
Iris-setosa     1.0000000 1.0000000 1.0000000
Iris-versicolor 0.8666667 1.0000000 0.9285714
Iris-virginica  1.0000000 0.8461538 0.9166667
```

References:

- R and Data Mining. Yangchan Zhao. Academic Press 2012. https://www.sciencedirect.com/book/9780123969637/r-and-data-mining

- Computing Classification Evaluation Metrics in R. *by Said Bleik, Shaheen Gauher, Data Scientists at Microsoft.* *http://blog.revolutionanalytics.com/2016/03/com_class_eval_metrics_r.html*