

CART Practice

Loading packages

```
if(!requireNamespace("caTools")) install.packages('caTools')

## Loading required namespace: caTools
if(!requireNamespace("tidyverse")) install.packages('tidyverse')

## Loading required namespace: tidyverse
if(!requireNamespace("caret")) install.packages('caret')

## Loading required namespace: caret
if(!requireNamespace("rpart")) install.packages('rpart')
if(!requireNamespace("rattle")) install.packages('rattle')

## Loading required namespace: rattle
library(caTools)
library(caret)

## Loading required package: ggplot2
## Loading required package: lattice
library(tidyverse)

## Warning: package 'tidyr' was built under R version 4.2.3
## Warning: package 'readr' was built under R version 4.2.3
## Warning: package 'dplyr' was built under R version 4.2.3
## Warning: package 'stringr' was built under R version 4.2.3

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr    1.5.1
## v lubridate  1.9.3      v tibble     3.2.1
## v purrr      1.0.2      v tidyr      1.3.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(rpart)
library(rattle)

## Loading required package: bitops
## Rattle: A free graphical interface for data science with R.
```

```
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

Reading data

1. Exclude Name, Ticket, Cabin
2. Omit missing value in Age
3. Survived and Pclass as factor

```
data <- read.csv("Titanic.csv")
data <- subset(data, select=-c(Name,Ticket,Cabin))
data <- subset(data,!is.na(Age))
cat("There are", nrow(data), "passengers left.")
```

```
## There are 714 passengers left.
```

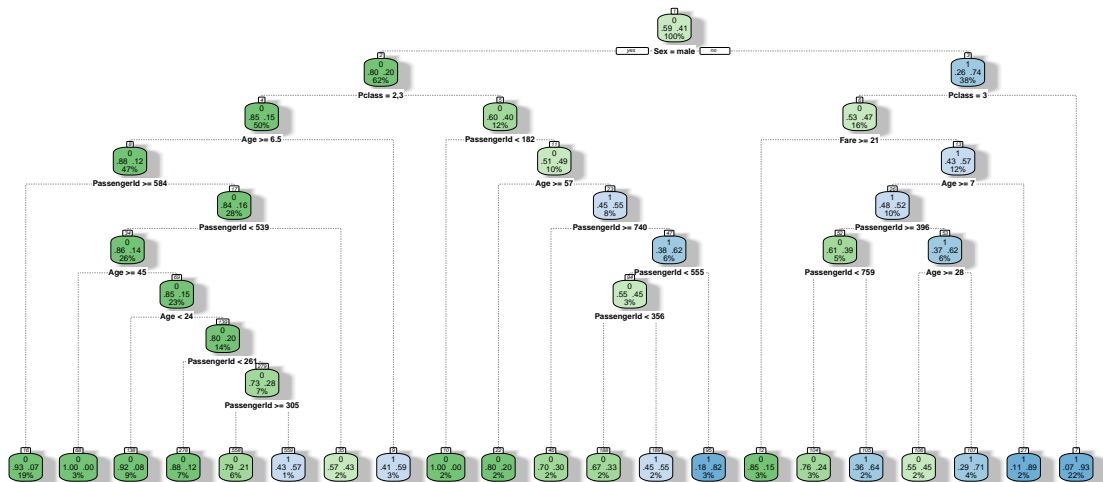
```
data$Survived <- as.factor(data$Survived)
data$Pclass <- as.factor(data$Pclass)
```

Split data

```
set.seed(123)
training_samples <- data$Survived %>%
  createDataPartition(p=0.8,list=FALSE)
train.data <- data[training_samples,]
test.data <- data[-training_samples,]
```

Fully grown tree

```
model <- rpart(Survived~., data=train.data, control=rpart.control(cp=0))
fancyRpartPlot(model)
```



Confusion matrix

```
pred <- predict(model, newdata=test.data, type = "class")
pred <- ifelse(pred==1, 'predict_1','predict_0')
table(pred,test.data$Survived)
```

```
##
## pred      0  1
## predict_0 71 11
## predict_1 13 47
```

Sensitivity

```
47/(47+11)
```

```
## [1] 0.8103448
```

Specificity

```
71/(71+13)
```

```
## [1] 0.8452381
```

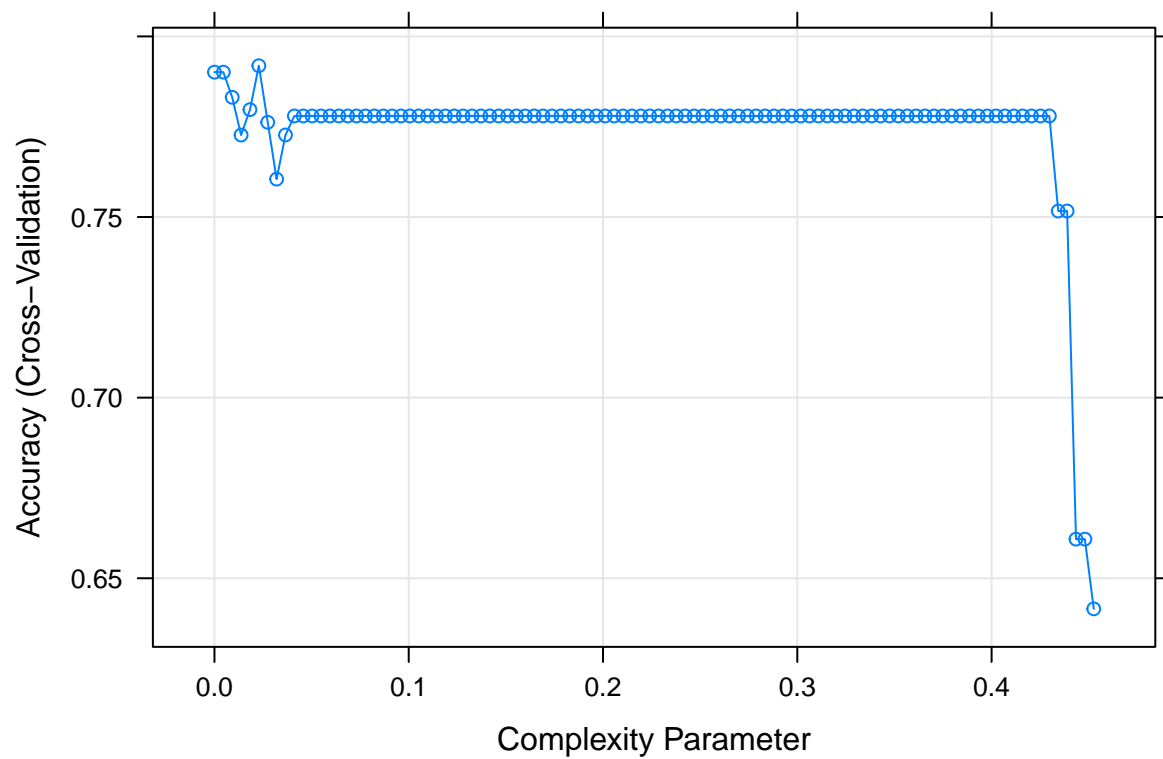
Accuracy

```
(47+71)/(47+71+11+13)
```

```
## [1] 0.8309859
```

Prune the tree with 10-fold cross-validation

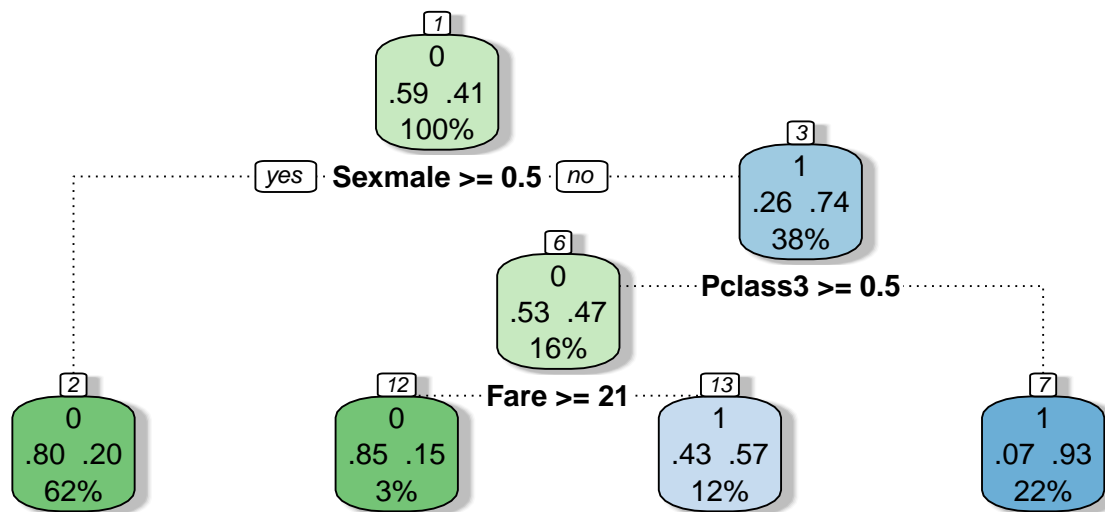
```
set.seed(123)
model2 <- train(Survived~., data=train.data, method="rpart",trControl=trainControl("cv",number=10),tune
plot(model2)
```



```
model2$bestTune
```

```
##          cp
## 6 0.02285789
```

```
fancyRpartPlot(model2$finalModel)
```



Rattle 2024-Mar-21 15:26:53 chaeunshin

Prediction and confusion matrix

```
pred <- predict(model2, newdata=test.data)
pred <- ifelse(pred==1, 'survived', 'dead')
table(pred, test.data$Survived)
```

```
##
## pred      0  1
##  dead    79 22
## survived  5 36
```

Sensitivity

```
36/(36+22)
```

```
## [1] 0.6206897
```

Specificity

```
79/(79+5)
```

```
## [1] 0.9404762
```

Accuracy

```
(79+36)/(79+22+5+36)
```

```
## [1] 0.8098592
```