Quiz 6

Chaeeun Shin

03/26/2024

Installing packages

```
library(MASS)
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 ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(caTools)
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.
```

```
library(openxlsx)
```

#1. # Reading data, deleting missing values

```
setwd("/Users/chaeeunshin/Desktop/AMS 580")
data <- read.csv("GreatUnknown.csv")
cat("There were originally",nrow(data), "cases in the data.","\n")
```

```
## There were originally 4601 cases in the data.
```

```
data <- na.omit(data)
data$y <- as.factor(data$y)
cat("There are", nrow(data), "cases left.","\n")
```

```
## There are 4601 cases left.
```

Spliting training and testing

```
set.seed(456)
training_samples <- data$y %>%
  createDataPartition(p=0.75,list=FALSE)
train.data <- data[training_samples,]
test.data <- data[-training_samples,]
nrow(train.data)
```

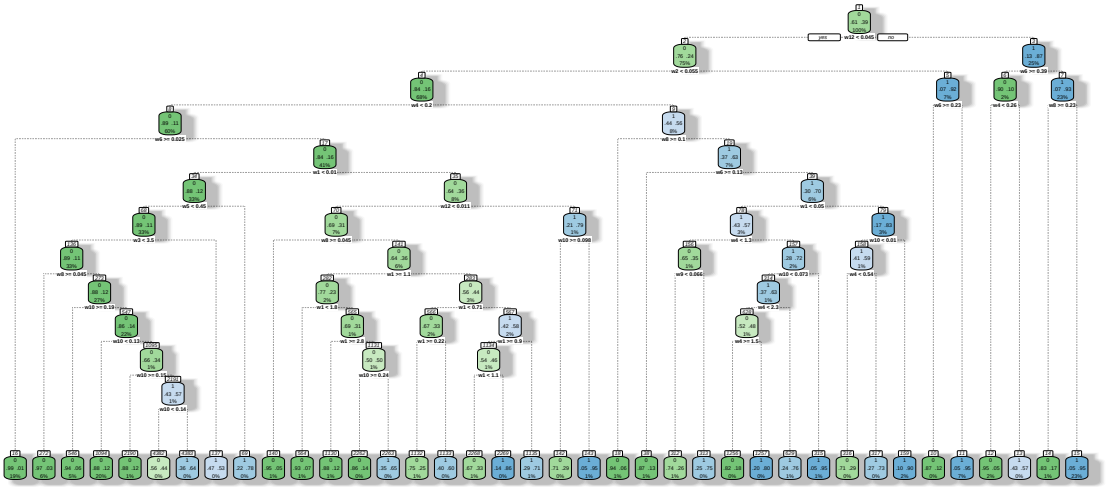
```
## [1] 3451
```

```
nrow(test.data)
```

```
## [1] 1150
```

#2. # Fully grown tree, drawing the tree plot

```
model <- rpart(y~., data=train.data,control=rpart.control(cp=0), method="class")
fancyRpartPlot(model)
```



Rattle 2024−Mar−26 02:32:45 chaeeunshin

Prediction: Confusion matrix, sensitivity, specificity, accuracy

```
pred <- predict(model, newdata=test.data, type="class")
fulltreepred <- ifelse(pred==1, 1,0)
fulltreeconfusion <- table(pred,test.data$y)
print(fulltreeconfusion)
```

```
##
## pred   0   1
##    0 653  63
##    1  44 390
```

```
cat("Sensitivity: ", fulltreeconfusion[2,2]/(fulltreeconfusion[2,1]+fulltreeconfusion[2,2]),"\n")
```

```
## Sensitivity:  0.8986175
```

```
cat("Specificity: ", fulltreeconfusion[1,1]/(fulltreeconfusion[1,1]+fulltreeconfusion[1,2]),"\n")
```

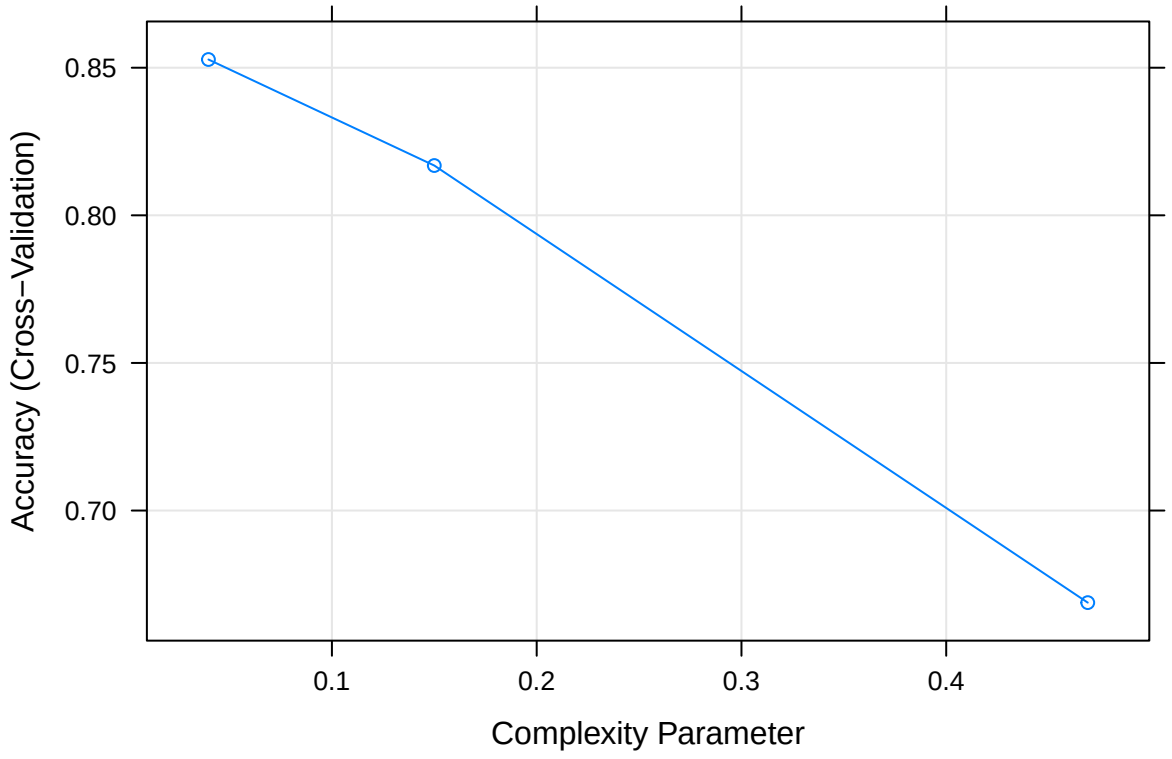
```
## Specificity:  0.9120112
```

```
cat("Accuracy: ", (fulltreeconfusion[1,1]+fulltreeconfusion[2,2])/(fulltreeconfusion[1,1]+fulltreeconfus
```

```
## Accuracy:  0.9069565
```

#3. # Prune the tree with 10-fold cross-validation

```
set.seed(456)
model2 <- train(y~., data=train.data, method="rpart", trControl = trainControl(method="cv",number=10))
plot(model2)
```

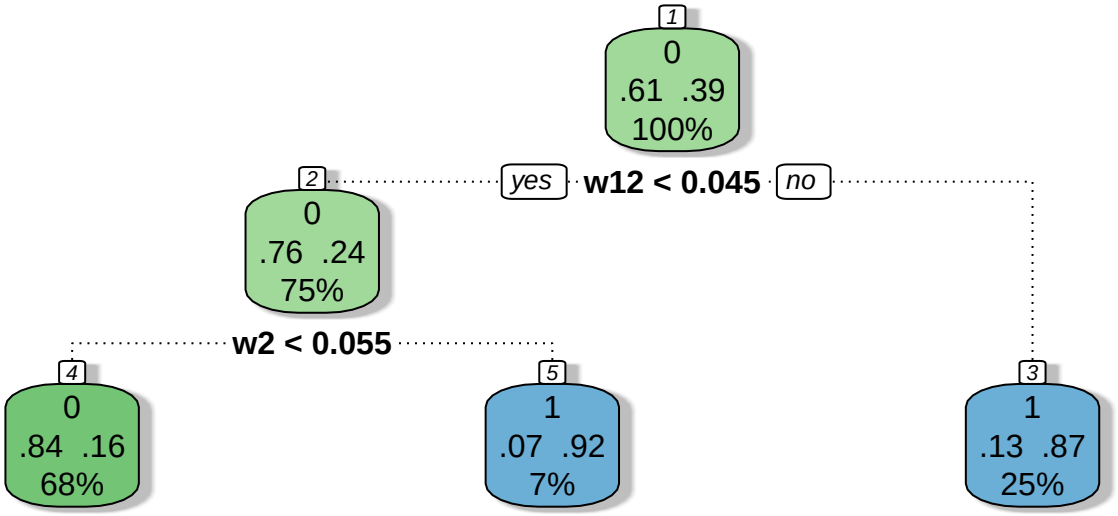



```
model2$bestTune
```

```
##           cp
```

```
## 1 0.03970588
```

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



Rattle 2024−Mar−26 02:32:46 chaeeunshin

#4.

```
pred <- predict(model2, newdata=test.data)
prunedtreepred <- ifelse(pred==1, 1,0)
prunedcm <- table(prunedtreepred, test.data$y)
print(prunedcm)
```

```
##
## prunedtreepred   0   1
##              0 643 115
##              1  54 338
```

```
cat("Sensitivity: ", prunedcm[2,2]/(prunedcm[2,2]+prunedcm[2,1]),"\n")
```

```
## Sensitivity:  0.8622449
```

```
cat("Specificity: ", (prunedcm[1,1]/(prunedcm[1,1]+prunedcm[1,2])),"\n")
```

```
## Specificity:  0.848285
```

```
cat("Accuracy: ", ((prunedcm[1,1]+prunedcm[2,2])/(prunedcm[1,1]+prunedcm[1,2]+prunedcm[2,1]+prunedcm[2,
```

```
## Accuracy:  0.8530435
```

#5.

```
logmodel <- glm(y~.,data=train.data,family="binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
probabilities <- logmodel %>% predict(test.data, type = "response")
logisticpred <- ifelse(probabilities>0.5, "1", "0")
logisticcm <- table(logisticpred,test.data$y)
print(logisticcm)
```

```
##
```

```
## logisticpred   0   1
##            0 666 101
##            1  31 352
```

```
cat("Sensitivity: ", (logisticcm[2,2]/(logisticcm[2,1]+logisticcm[2,2])),"\n")
```

```
## Sensitivity:  0.9190601
```

```
cat("Specificity: ", logisticcm[1,1]/(logisticcm[1,2]+logisticcm[1,1]),"\n")
```

```
## Specificity:  0.8683181
```

```
cat("Accuracy: ", (logisticcm[1,1]+logisticcm[2,2])/(logisticcm[1,1]+logisticcm[1,2]+logisticcm[2,1]+log
```

```
## Accuracy:  0.8852174
```

#6.

```
output <- data.frame(Full_tree_prediction = fulltreepred, Pruned_tree_prediction = prunedtreepred, Logis
write.xlsx(output, "Prediction output.xlsx")
```

#Generating majority vote

```
majorityvote <- function(x) {
  return(names(sort(table(x),decreasing=TRUE))[1])
}
final_pred <- apply(output,1,majorityvote)
```

Confusion matrix, sensifity, specifity, accuracy

```
finalconfusion <- table(final_pred, test.data$y)
print(finalconfusion)
```

```
##
## final_pred   0   1
##          0 664  92
##          1  33 361
```

```
cat("Sensitivity: ", finalconfusion[2,2]/(finalconfusion[2,2]+finalconfusion[2,1]),"\n")
```

```
## Sensitivity:  0.9162437
```

```
cat("Specificity: ",finalconfusion[1,1]/(finalconfusion[1,1]+finalconfusion[1,2]),"\n")
```

```
## Specificity:  0.8783069
```

```
cat("Accuracy: ", (finalconfusion[1,1]+finalconfusion[2,2])/(finalconfusion[1,1]+finalconfusion[1,2]+fi
```

```
## Accuracy:  0.8913043
```