Quiz 6

Srinivasa Phani Madhav Marupudi

2024-03-25

Loading packages and data

```
# if (!requireNamespace("caTools")) install.packages('caTools')
# if (!requireNamespace("tidyverse")) install.packages('tidyverse')
# if (!requireNamespace("caret")) install.packages('caret')
# if (!requireNamespace("rpart")) install.packages('rpart')
# if (!requireNamespace("rattle")) install.packages('rattle')
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.3.3
```

```
library(tidyverse)
```

```
## -- 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   3.5.0     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()
## 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(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.3.3
```

```
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 4.3.3
```

```
## 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.
```

q1

```
df <- read.csv("C:/Users/MSP/Downloads/GreatUnknown.csv")

df <- na.omit(df)
cases_left <- nrow(df)
cat("Number of cases left after cleaning:", cases_left, "\n")
```

```
## Number of cases left after cleaning: 4601
```

```
df$y = as.factor(df$y)

set.seed(456)

# train_indices <- sample(nrow(cleaned_df), 0.75 * nrow(cleaned_df))
train_indices = df$y %>% createDataPartition(p=0.75, list = F)
train_data <- df[train_indices, ]
test_data <- df[-train_indices, ]

# write.csv(train_data, "train_data.csv", row.names = FALSE)
# write.csv(test_data, "test_data.csv", row.names = FALSE)
```

q2

```
# train_data <- read.csv("train_data.csv")
tree_model <- rpart(y ~ ., data = train_data, method = "class")
# test_data <- read.csv("test_data.csv")

predictions <- predict(tree_model, test_data, type = "class")

conf_matrix <- table(predictions, test_data$y)

sensitivity <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
specificity <- conf_matrix[1, 1] / sum(conf_matrix[1, ])
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

print("Confusion Matrix:")
```

```
## [1] "Confusion Matrix:"
```

```
print(conf_matrix)
```

```
##
## predictions   0   1
##           0 647  62
##           1  50 391
```

```
print(paste("Sensitivity:", sensitivity))
```

```
## [1] "Sensitivity: 0.886621315192744"
```

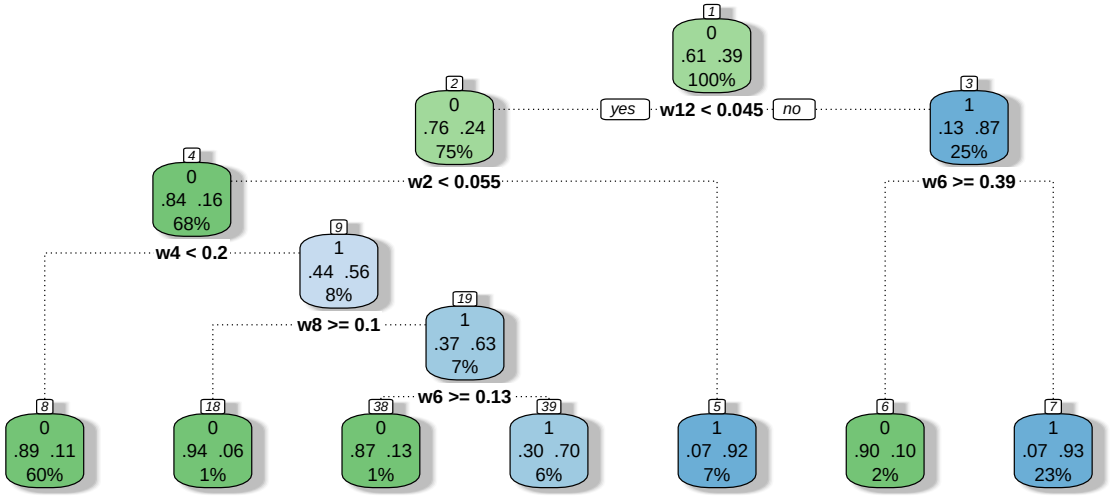
```
print(paste("Specificity:", specificity))
```

```
## [1] "Specificity: 0.912552891396333"
```

```
print(paste("Overall Accuracy:", accuracy))
```

```
## [1] "Overall Accuracy: 0.902608695652174"
```

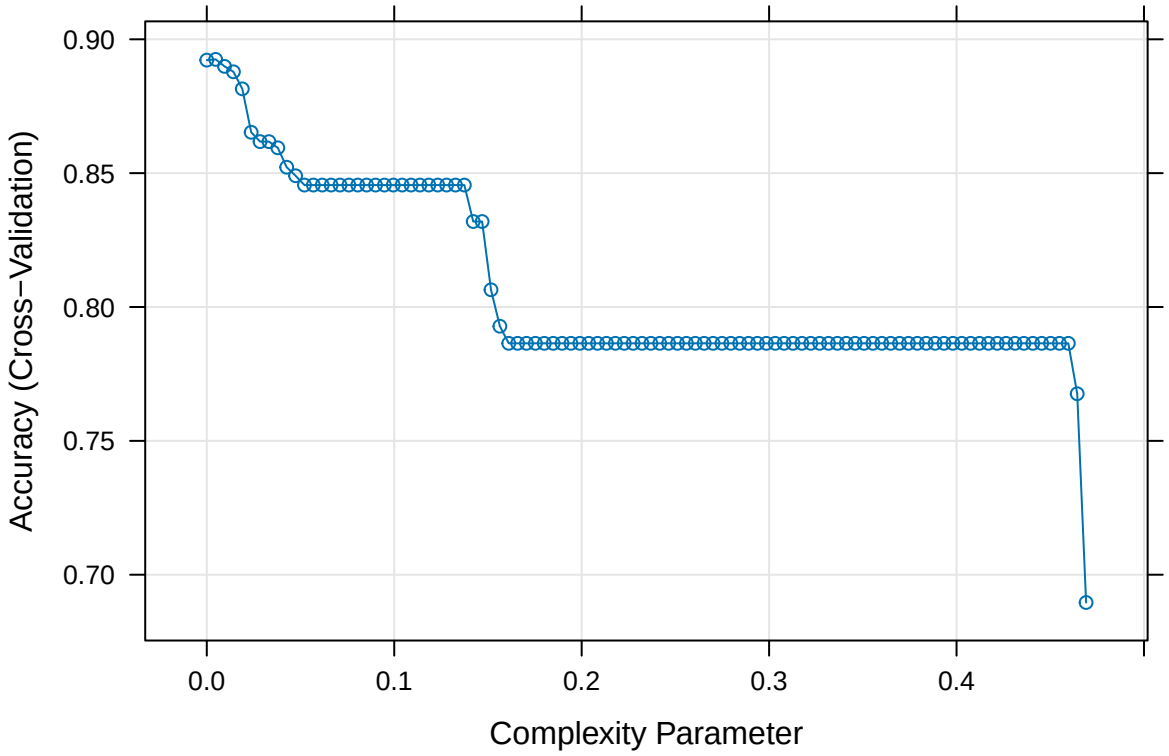
```
fancyRpartPlot(tree_model)
```



Rattle 2024−Mar−25 16:44:03 MSP

q3

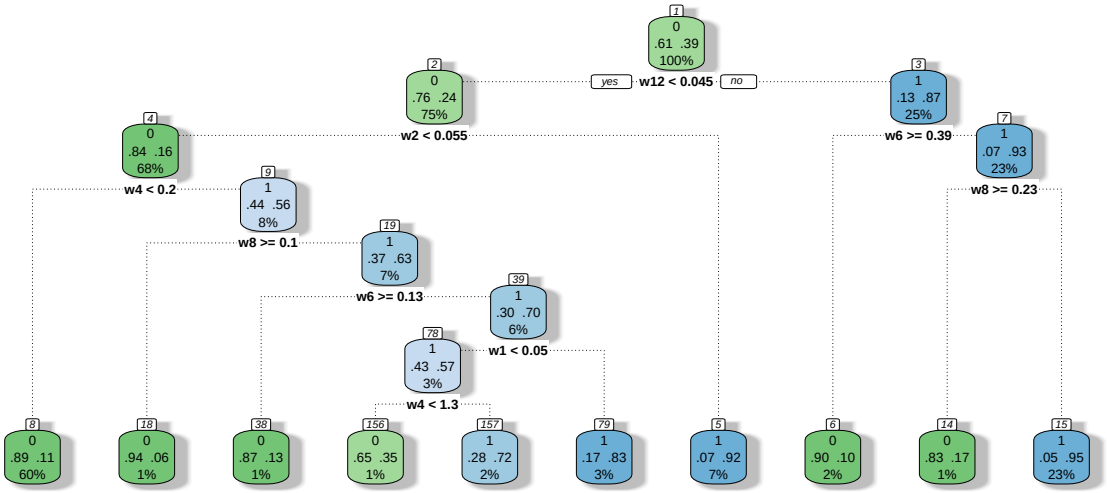
```
# set.seed(456)
model2 <- train(
  y ~., data = train_data, method = "rpart",
  trControl = trainControl("cv", number = 10),
  tuneLength = 100)
plot(model2)
```




```
model2$bestTune
```

```
##           cp
## 2 0.004738562
```

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



Rattle 2024−Mar−25 16:44:07 MSP

q4

```
predictions_pruned <- predict(model2, test_data)
conf_matrix_pruned <- table(predictions_pruned, test_data$y)

sensitivity_pruned <- conf_matrix_pruned[2, 2] / sum(conf_matrix_pruned[2, ])
specificity_pruned <- conf_matrix_pruned[1, 1] / sum(conf_matrix_pruned[1, ])
accuracy_pruned <- sum(diag(conf_matrix_pruned)) / sum(conf_matrix_pruned)

print("Confusion Matrix (Pruned Tree):")
```

```
## [1] "Confusion Matrix (Pruned Tree):"
```

```
print(conf_matrix_pruned)
```

```
##
## predictions_pruned   0   1
##                  0 662  72
##                  1  35 381
```

```
print(paste("Sensitivity (Pruned Tree):", sensitivity_pruned))
```

```
## [1] "Sensitivity (Pruned Tree): 0.915865384615385"
```

```
print(paste("Specificity (Pruned Tree):", specificity_pruned))
```

```
## [1] "Specificity (Pruned Tree): 0.901907356948229"
```

```r
print(paste("Overall Accuracy (Pruned Tree):", accuracy_pruned))
```

```
## [1] "Overall Accuracy (Pruned Tree): 0.90695652173913"
```

q5

```r
library(glm2)

logit_model <- glm(y ~ ., data = train_data, family = binomial)
```

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

```r
predictions_logit <- predict(logit_model, newdata = test_data, type = "response")
predictions_logit <- ifelse(predictions_logit > 0.5, 1, 0)
conf_matrix_logit <- table(predictions_logit, test_data$y)

sensitivity_logit <- conf_matrix_logit[2, 2] / sum(conf_matrix_logit[2, ])
specificity_logit <- conf_matrix_logit[1, 1] / sum(conf_matrix_logit[1, ])
accuracy_logit <- sum(diag(conf_matrix_logit)) / sum(conf_matrix_logit)

print("Confusion Matrix (Logistic Regression Model):")
```

```
## [1] "Confusion Matrix (Logistic Regression Model):"
```

```r
print(conf_matrix_logit)
```

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

```r
print(paste("Sensitivity (Logistic Regression Model):", sensitivity_logit))
```

```
## [1] "Sensitivity (Logistic Regression Model): 0.919060052219321"
```

```r
print(paste("Specificity (Logistic Regression Model):", specificity_logit))
```

```
## [1] "Specificity (Logistic Regression Model): 0.868318122555411"
```

```r
print(paste("Overall Accuracy (Logistic Regression Model):", accuracy_logit))
```

```
## [1] "Overall Accuracy (Logistic Regression Model): 0.885217391304348"
```

q6

```r
combined_preds <- data.frame(test_data, predictions, predictions_pruned, predictions_logit)
write.csv(train_data, "combined_predictions.csv")
```

```r
most_predicted <- apply(combined_preds[, c("predictions", "predictions_pruned", "predictions_logit")], 1,
    class_counts <- table(x)
    names(class_counts)[which.max(class_counts)]
  })

conf_matrix <- table(most_predicted, combined_preds$y)
conf_matrix
```

```
##
## most_predicted   0   1
##              0 659  71
##              1  38 382
```

```
sensitivity <- conf_matrix[2, 2] / sum(conf_matrix[2, ])
specificity <- conf_matrix[1, 1] / sum(conf_matrix[1, ])
accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

print(paste("sensitivity: ", sensitivity))
```

```
## [1] "sensitivity:  0.90952380952381"
```

```
print(paste("specificity: ", specificity))
```

```
## [1] "specificity:  0.902739726027397"
```

```
print(paste("accuracy: ", accuracy))
```

```
## [1] "accuracy:  0.905217391304348"
```