

# hw13

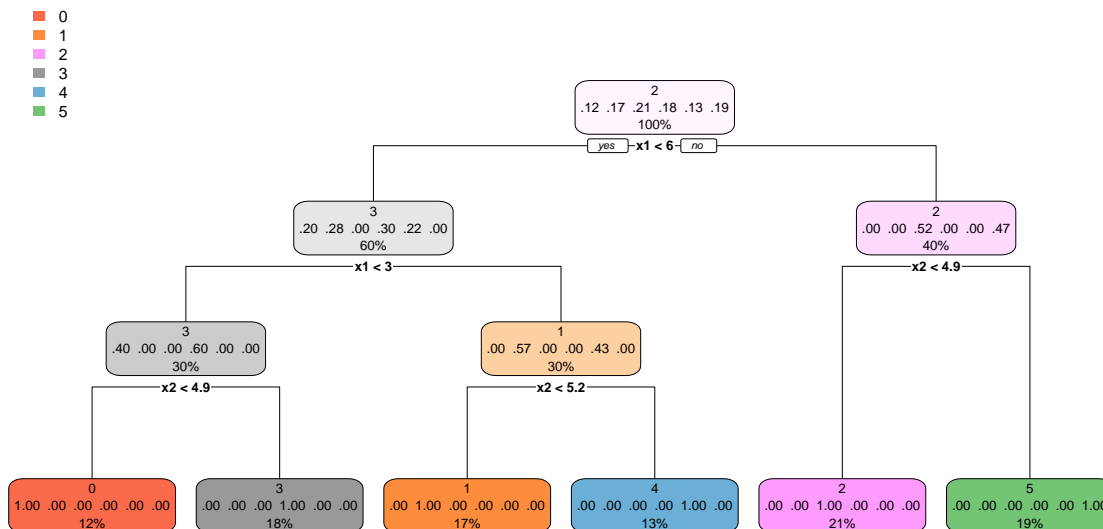
Getong Zhong

2023-05-01

1

```
library(rpart)
library(rpart.plot)
set.seed(123)
n <- 100
x1 <- runif(n, 0, 10)
x2 <- runif(n, 0, 10)
assign_region <- function(x1, x2) {
  if (x1 < 3 & x2 < 5) return(0)
  if (x1 >= 3 & x1 < 6 & x2 < 5) return(1)
  if (x1 >= 6 & x2 < 5) return(2)
  if (x1 < 3 & x2 >= 5) return(3)
  if (x1 >= 3 & x1 < 6 & x2 >= 5) return(4)
  if (x1 >= 6 & x2 >= 5) return(5)
}
y <- mapply(assign_region, x1, x2)
df <- data.frame(x1, x2, y)
tree <- rpart(y ~ x1 + x2, data = df, method = "class")

rpart.plot(tree)
```

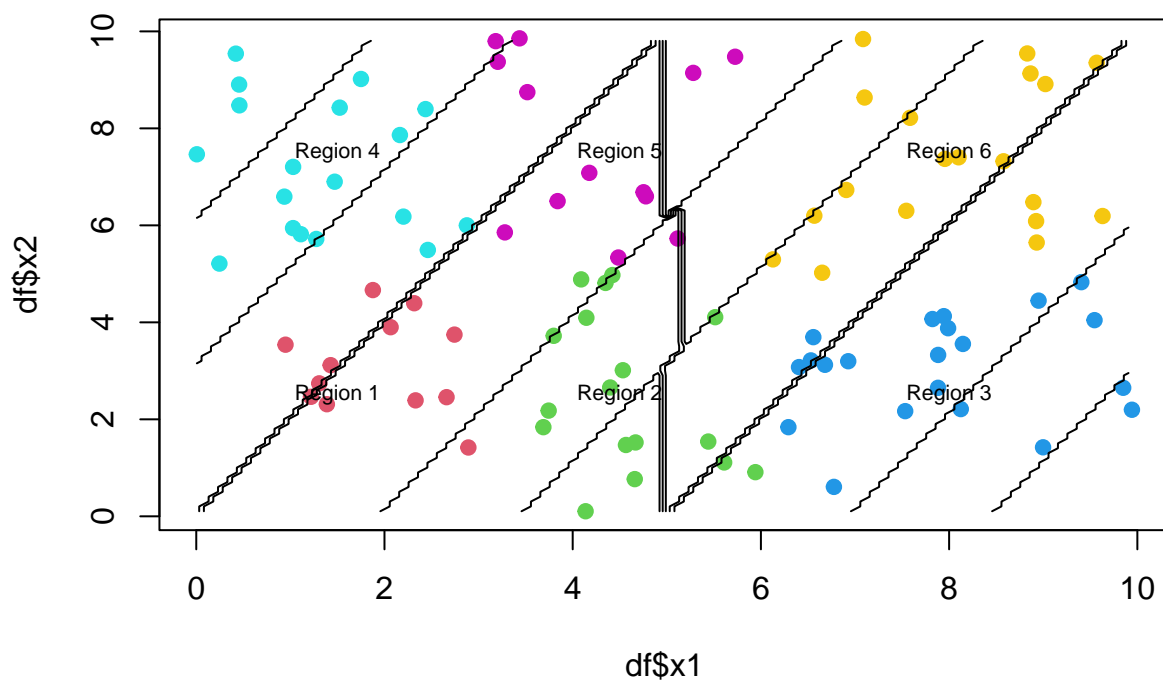


```
grid_size <- 0.1
grid_x1 <- seq(min(df$x1), max(df$x1), by = grid_size)
grid_x2 <- seq(min(df$x2), max(df$x2), by = grid_size)
grid_df <- expand.grid(x1 = grid_x1, x2 = grid_x2)

# Predict the grid points using the decision tree
grid_df$y_pred <- predict(tree, grid_df, type = "class")
# Create a matrix of the predicted values for contour plot
pred_matrix <- matrix(grid_df$y_pred, nrow = length(grid_x2), ncol = length(grid_x1))

# Add the contour plot (decision boundary)
plot(df$x1, df$x2, col = y + 2, pch = 19)

# Add the contour plot (decision boundaries)
contour(grid_x1, grid_x2, t(pred_matrix), levels = c(0.5, 1.5, 2.5, 3.5, 4.5), drawlabels = FALSE, add = FALSE)
region_labels <- c("Region 1", "Region 2", "Region 3", "Region 4", "Region 5", "Region 6")
region_centers <- data.frame(x1 = c(1.5, 4.5, 8, 1.5, 4.5, 8), x2 = c(2.5, 2.5, 2.5, 7.5, 7.5, 7.5))
text(region_centers$x1, region_centers$x2, labels = region_labels, cex = 0.7, col = "black")
```



## 6 First the algorithm starts with the whole data set as the root node and repeatedly select the best split point to minimize the residuals. And then, the data partitioned based on the selected split points, and repeat this process for each child node until the end of the iteration, usually known as when the maximum depth are met. When we met the end of the iteration, terminal nodes are created, meanwhile, the mean of the target variable within the node is assigned to each child node.