College Student Placement

Claudia Clinchard

2025-11-06

```
library(rpart)
library(rpart.plot)
library(dplyr)
```

```
## 
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
## 
##     filter, lag
```

```
## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(purrr)
```

Inspect the data

```
colnames(college)
```

```
##  [1] "College_ID"             "IQ"                     "Prev_Sem_Result"       

##  [4] "CGPA"                   "Academic_Performance"   "Internship_Experie
nce"
##  [7] "Extra_Curricular_Score" "Communication_Skills"   "Projects_Completed
"
## [10] "Placement"
```

```
head(college)
```

```
##   College_ID  IQ Prev_Sem_Result CGPA Academic_Performance
## 1    CLG0030 107            6.61 6.28                    8
## 2    CLG0061  97            5.52 5.37                    8
## 3    CLG0036 109            5.36 5.83                    9
## 4    CLG0055 122            5.47 5.75                    6
## 5    CLG0004  96            7.91 7.69                    7
## 6    CLG0015  96            5.26 5.32                    7
##   Internship_Experience Extra_Curricular_Score Communication_Skills
## 1                    No                      8                    8
## 2                    No                      7                    8
## 3                    No                      3                    1
## 4                   Yes                      1                    6
## 5                    No                      8                   10
## 6                    No                      5                    8
##   Projects_Completed Placement
## 1                  4        No
## 2                  0        No
## 3                  1        No
## 4                  1        No
## 5                  2        No
## 6                  0        No
```

```r
str(college)
```

```
## 'data.frame':    10000 obs. of  10 variables:
##  $ College_ID            : chr  "CLG0030" "CLG0061" "CLG0036" "CLG0055"
 ...
##  $ IQ                    : int  107 97 109 122 96 96 123 111 92 108 ...
##  $ Prev_Sem_Result       : num  6.61 5.52 5.36 5.47 7.91 5.26 6.68 8.77 6.
47 8.82 ...
##  $ CGPA                  : num  6.28 5.37 5.83 5.75 7.69 5.32 6.58 8.76 6.
33 8.6 ...
##  $ Academic_Performance  : int  8 8 9 6 7 7 5 7 9 4 ...
##  $ Internship_Experience : chr  "No" "No" "No" "Yes" ...
##  $ Extra_Curricular_Score: int  8 7 3 1 8 5 7 3 7 5 ...
##  $ Communication_Skills  : int  8 8 1 6 10 8 8 1 8 9 ...
##  $ Projects_Completed    : int  4 0 1 1 2 0 2 2 5 1 ...
##  $ Placement             : chr  "No" "No" "No" "No" ...
```

```r
college <- college %>%
  mutate(across(c(Academic_Performance, IQ, Extra_Curricular_Score,
                   Communication_Skills, Projects_Completed),
                as.numeric)) %>%
  mutate(across(c(Internship_Experience, Placement),
                as.factor))
str(college)
```

```
## 'data.frame':    10000 obs. of  10 variables:
##  $ College_ID            : chr  "CLG0030" "CLG0061" "CLG0036" "CLG0055"
 ...
##  $ IQ                    : num  107 97 109 122 96 96 123 111 92 108 ...
##  $ Prev_Sem_Result       : num  6.61 5.52 5.36 5.47 7.91 5.26 6.68 8.77 6.
47 8.82 ...
##  $ CGPA                  : num  6.28 5.37 5.83 5.75 7.69 5.32 6.58 8.76 6.
33 8.6 ...
##  $ Academic_Performance  : num  8 8 9 6 7 7 5 7 9 4 ...
##  $ Internship_Experience : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 1 1
1 1 ...
##  $ Extra_Curricular_Score: num  8 7 3 1 8 5 7 3 7 5 ...
##  $ Communication_Skills  : num  8 8 1 6 10 8 8 1 8 9 ...
##  $ Projects_Completed    : num  4 0 1 1 2 0 2 2 5 1 ...
##  $ Placement             : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 2
1 1 ...
```

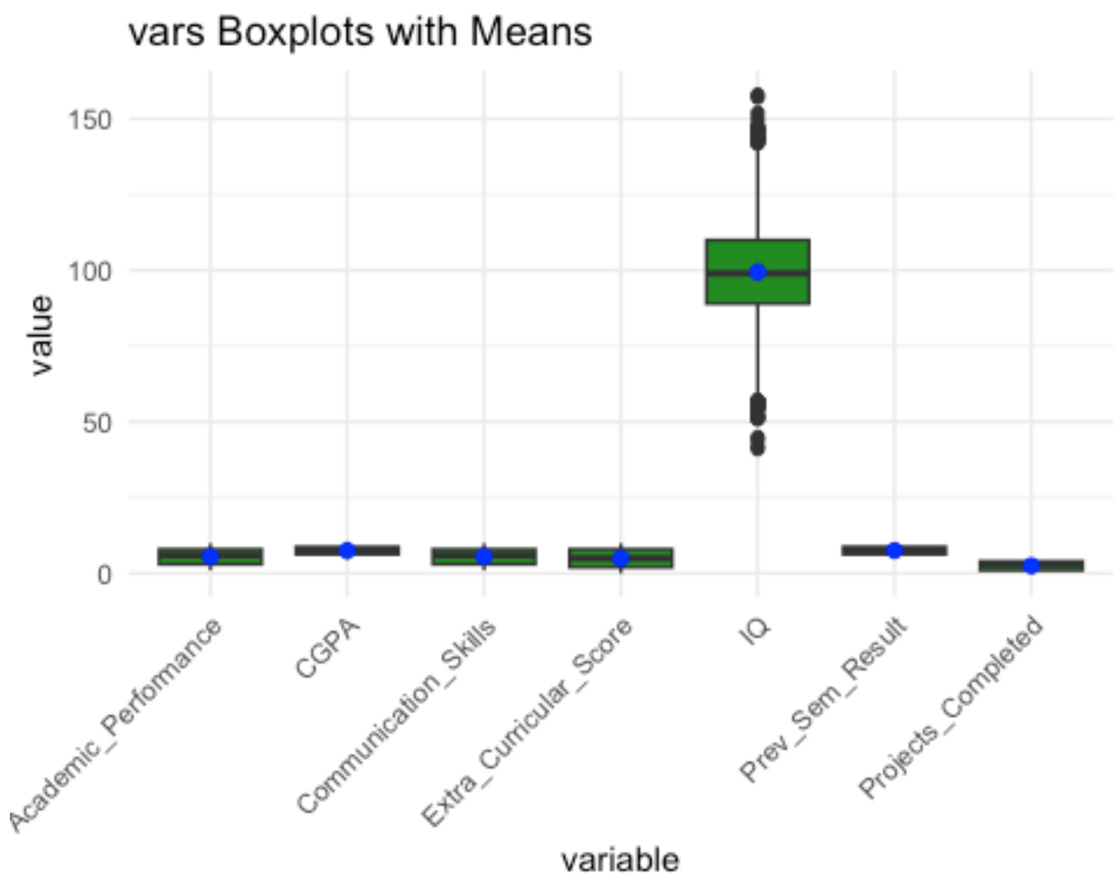
```r
table(college$Placement)
```

```
##
##   No  Yes
## 8341 1659
```

Some exploratory data analysis (EDA)

```
variables <- list(
  vars = c("IQ", "Prev_Sem_Result", "CGPA", "Academic_Performance",
            "Extra_Curricular_Score", "Communication_Skills", "Projects_Compl
eted")
)


# Function to plot and save histograms
plot_hist_vars <- function(df, cols, vars, path = ".") {
  n <- length(cols)
  nrow <- ceiling(n / 4)

  png(filename = file.path(path, paste0(vars, "_histograms.png")),
      width = 1600, height = 800)

  par(mfrow = c(nrow, 4), mar = c(4, 4, 2, 1))

  for (col in cols) {
    hist(df[[col]],
         main = col,
         xlab = "Value",
         col = "forestgreen",
         border = "white")
  }
   title(main = paste(vars, "Histograms"), outer = TRUE, line = -1.5)
  dev.off()
}

plot_box_vars <- function(df, cols, vars, path = ".") {
  df_long <- df %>%
    dplyr::select(all_of(cols)) %>%
    tidyr::pivot_longer(cols = everything(), names_to = "variable", values_to
 = "value")

   p <- ggplot(df_long, aes(x = variable, y = value)) +
    geom_boxplot(fill = "forestgreen") +
    stat_summary(fun = mean, geom = "point", shape = 20, size = 3, color = "b
lue") +
    theme_minimal() +
    labs(title = paste(vars, "Boxplots with Means")) +
         theme(axis.text.x = element_text(angle = 45, hjust = 1))

  ggsave(filename = file.path(path, paste0(vars, "_boxplot.png")),
```

```r
         plot = p, width = 10, height = 6)

  return(p)
}

# Directory to save plots
output_dir <- "plots"
if (!dir.exists(output_dir)) dir.create(output_dir)

# Loop through all variable groups
vars_boxplots <- list()
for (vars in names(variables)) {
  cols_exist <- variables[[vars]][variables[[vars]] %in% names(college)]

  if (length(cols_exist) == 0) {
    warning(paste("No valid columns found for", vars))
    next
  }

  plot_hist_vars(college, cols_exist, vars, path = output_dir)
  vars_boxplots[[vars]] <- plot_box_vars(college, cols_exist, vars, path = ou
tput_dir)
}
#loop through all to print boxplots
walk(vars_boxplots, print)
```



Build the initial classification tree

```
tree1 <- rpart(Placement ~ IQ + Prev_Sem_Result + CGPA + Academic_Performance
 + Internship_Experience +
                 Extra_Curricular_Score + Communication_Skills + Projects_Com
pleted, data = college,
               control = rpart.control(cp=.01), method = "class")
printcp(tree1)

## 
## Classification tree:
## rpart(formula = Placement ~ IQ + Prev_Sem_Result + CGPA + Academic_Perform
ance + 
##     Internship_Experience + Extra_Curricular_Score + Communication_Skills 
+ 
##     Projects_Completed, data = college, method = "class", control = rpart.
control(cp = 0.01))
## 
## Variables actually used in tree construction:
## [1] CGPA                 Communication_Skills IQ                  
## [4] Projects_Completed  
## 
## Root node error: 1659/10000 = 0.1659
```

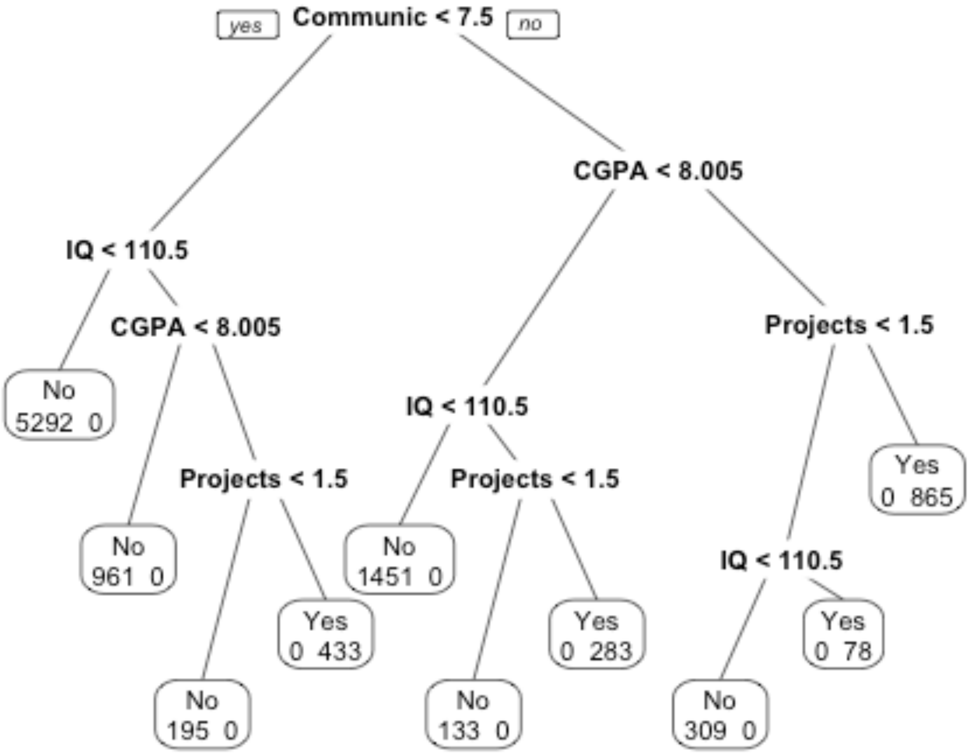
```
##
## n= 10000
##
##          CP nsplit rel error   xerror      xstd
## 1 0.191079      0  1.000000 1.000000 0.0224226
## 2 0.139241      2  0.617842 0.617842 0.0182824
## 3 0.090416      3  0.478602 0.478602 0.0162967
## 4 0.080169      4  0.388186 0.404461 0.0150811
## 5 0.071730      5  0.308017 0.308017 0.0132732
## 6 0.047016      8  0.047016 0.047016 0.0053027
## 7 0.010000      9  0.000000 0.000000 0.0000000
```

Identify the best cp value to use

```
best <- tree1$cptable[which.min(tree1$cptable[,"xerror"]),"CP"]
best
```

```
## [1] 0.01
```

```
pruned_tree <- prune(tree1, cp=best)

prp(pruned_tree,
    faclen=0, #use full names for factor labels
    extra=1, #display number of obs. for each terminal node
    roundint=F, #don't round to integers in output
    digits=5) #display 5 decimal places in output
```



We can see that someone with communication above than 7.5, a CGPA above 8.005, less than 1.5 project, and an IQ greater than 110.5 would receive a placement.

See an example for someone with IQ of 120, Previous Semester Result of 8, GPA of 3.75, Academic Performance of 6, Yes for Internship, 5 for Extracurricular score, 7 for communication skills, and 3 projects completed; we know from the diagram that they will not receive a placement, but let’s check

```
new <- data.frame(
  IQ = 120,
  Prev_Sem_Result = 8,
  CGPA = 3.75,
  Academic_Performance = 6,
  Internship_Experience = factor("Yes", levels = c("No","Yes")),
  Extra_Curricular_Score = 5,
  Communication_Skills = 7,
  Projects_Completed = 3
)
predict(pruned_tree, newdata=new, type = "class")
```

```
##  1
## No
## Levels: No Yes
```