Prezentacja końcowa

Analiza danych o uczniach

Paulina Bannert,
Natalia Bercz ,
Dariusz Sudół,
Piotr Mrozik,
Monika Wyźnikiewicz

Dane, z jakimi pracowaliśmy

- dane z dwóch portugalskich szkół,
- dane dotyczące uczniów, którzy uczęszczają na rozszerzoną matematykę i/lub portugalski,
- dane o płci, wieku, rozmiarze rodziny, czasu podróży do szkoły, ilości wolnego czasu, itp. tych uczniów

Możliwości aplikacji, które stworzyliśmy

- przeglądanie danych w tabeli, według wybranych przez użytkownika cech uczniów i szkół, do jakich chodzą,
- przeglądanie wykresów, na których zwizualizowano informacje o uczniach oraz różne zależności między ich cechami,
- prognozowanie oceny końcowej, według podanych przez użytkownika cech

Pobieranie i modyfikowanie danych

```
9  mat <- fread("student-mat.csv")
10 port <- fread("student-por.csv")
11
12 setkeyv(mat, c("school", "sex", "age", "address", "famsize", "Pstatus", "Medu", "Fedu", "Mjob", "Fjob",
13                "reason", "nursery", "internet", "traveltime", "romantic", "guardian", "famrel", "studytime",
14                "schoolsup", "famsup", "activities", "higher", "freetime", "goout",  "Dalc", "Walc", "health"))
15
16 setkeyv(port, c("school", "sex", "age", "address", "famsize", "Pstatus", "Medu", "Fedu", "Mjob", "Fjob",
17                 "reason", "nursery", "internet", "traveltime", "romantic", "guardian", "famrel", "studytime",
18                 "schoolsup", "famsup", "activities", "higher", "freetime", "goout",  "Dalc", "Walc", "health"))
19
20 both <- mat[port, nomatch = 0]
21 together <- merge(mat, port, all = TRUE)
22
23 setnames(together,
24          c("G1.x", "G2.x", "G3.x", "failures.x", "paid.x", "absences.x", "G1.y", "G2.y", "G3.y", "failures.y", "paid.y", "absences.y"),
25          c("math.G1", "math.G2", "math.G3", "math.failures", "math.paid", "math.absences", "port.G1", "port.G2", "port.G3",
26            "port.failures", "port.paid", "port.absences"))
```

Stworzenie modeli

```
28 model_mat <- glm(G3 ~ factor(school) + factor(sex) + age + factor(address) + factor(famsize) + factor(Pstatus) + factor(Medu) +
29                     factor(Fedu) + factor(Mjob) + factor(Fjob) + factor(reason) + factor(guardian) + factor(traveltime) +
30                     factor(studytime) + factor(failures) + factor(schoolsup) + factor(famsup) + factor(paid) + factor(activities) +
31                     factor(nursery) + factor(higher) + factor(internet) + factor(romantic) + factor(famrel) + factor(freetime) +
32                     factor(goout) + factor(Dalc) + factor(Walc) + factor(health) + absences, data = mat)
33
34 model_por <- glm(G3 ~ factor(school) + factor(sex) + age + factor(address) + factor(famsize) + factor(Pstatus) + factor(Medu) +
35                     factor(Fedu) + factor(Mjob) + factor(Fjob) + factor(reason) + factor(guardian) + factor(traveltime) +
36                     factor(studytime) + factor(failures) + factor(schoolsup) + factor(famsup) + factor(paid) + factor(activities) +
37                     factor(nursery) + factor(higher) + factor(internet) + factor(romantic) + factor(famrel) + factor(freetime) +
38                     factor(goout) + factor(Dalc) + factor(Walc) + factor(health) + absences, data = port)
```

UI - ogólny wygląd aplikacji

```
ui <- fluidPage(
    theme = shinytheme("sandstone"),
    titlePanel("Student Performance Data Set"),
    tabsetPanel(
        tabPanel("Data",
                 sidebarLayout(
                     sidebarPanel(
                         selectInput(
                             inputId = "school",
                             label = "Choose school:",
                             choices = c("GP", "MS", "Both")
                         ),
                         pickerInput(
                             inputId = "check_cols",
                             label = "Select columns",
                             choices = colnames(together),
                             selected = c("sex", "age", "address"),
                             options = list(
                                 `actions-box` = TRUE),
                             multiple = TRUE
                         ),
                         tableOutput("data_cols_info")
                     ),
                     mainPanel(
                         dataTableOutput("table")
                     )
                 )
        ),
```

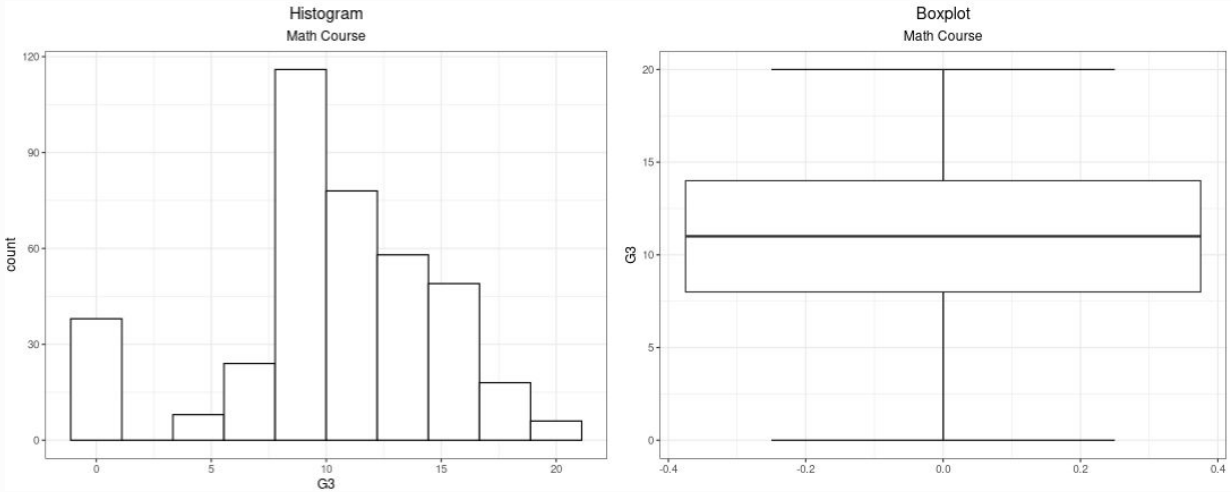
```
                switchInput(
                  label = "Show parameters",
                  inputId = "show_pred_param_switch",
                  value = FALSE
                ),
                conditionalPanel("input.show_pred_param_switch",
                                 tableOutput("pred_param1"),
                                 tableOutput("pred_param2"),
                                 tableOutput("pred_param3")
                ),
                actionButton(
                  inputId = "button_pred",
                  label = "Predict grade"
                ),
                textOutput("pred_output_mat"),
                textOutput("pred_output_por")
    ),
    tabPanel("About", uiOutput("about"))
  )
)
```

Panel Predykcji w UI

```
190         tabPanel("Predictions",
191                  fluidRow(
192                      column(2,
193                             pickerInput(inputId = "pred_school", label = "School", choices = sort(unique(mat$school)), selected = "GP"),
194                             pickerInput(inputId = "pred_sex", label = "Sex", choices = sort(unique(mat$sex)), selected = "F"),
195                             pickerInput(inputId = "pred_age", label = "Age", choices = 15:22, selected = 18),
196                             pickerInput(inputId = "pred_address", label = "Address", choices = sort(unique(mat$address)), selected = "U"),
197                             pickerInput(inputId = "pred_famsize", label = "Family Size", choices = sort(unique(mat$famsize)), selected = "GT3"),
198                             pickerInput(inputId = "pred_Pstatus", label = "Parent's Cohabitation Status", choices = sort(unique(mat$Pstatus)), selected = "A")
199                      ),
200                      column(2,
201                             pickerInput(inputId = "pred_Medu", label = "Mother's Education", choices = sort(unique(mat$Medu)), selected = 0),
202                             pickerInput(inputId = "pred_Fedu", label = "Father's Education", choices = sort(unique(mat$Fedu)), selected = 0),
203                             pickerInput(inputId = "pred_Mjob", label = "Mother's Job", choices = sort(unique(mat$Mjob)), selected = "at_home"),
204                             pickerInput(inputId = "pred_Fjob", label = "Father's Job", choices = sort(unique(mat$Fjob)), selected = "teacher"),
205                             pickerInput(inputId = "pred_reason", label = "Reason", choices = sort(unique(mat$reason)), selected = "course"),
206                             pickerInput(inputId = "pred_schoolsup", label = "School Educational Support", choices = sort(unique(mat$schoolsup)), selected = "yes")
207                      ),
```

Przykładowe wykresy w aplikacji

```
116             tabPanel("Hist+Box",
117                      sidebarLayout(
118                        sidebarPanel(
119                          radioGroupButtons(
120                            inputId = "histbox_subject",
121                            label = "Choose subject",
122                            choices = c("Math", "Portuguese"),
123                            justified = TRUE
124                          ),
125                          radioGroupButtons(
126                            inputId = "histbox_grade",
127                            label = "Choose grade",
128                            choices = c("G1", "G2", "G3"),
129                            justified = TRUE
130                          ),
131                          radioGroupButtons(
132                            inputId = "histogram_type",
133                            label = "Choose type",
134                            choices = c("density", "count"),
135                            justified = TRUE
136                          )
137                        ),
138                        mainPanel(
139                          splitLayout(cellWidths = c("50%", "50%"), plotOutput("histogram"), plotOutput("boxplot"))
140                        )
141                      )
142             ),
```



Serwer

```r
server <- function(input, output) {
  # Data
  output$table <- renderDataTable(
    together %>%
      { if (input$school != "Both") filter(., school == input$school) else . } %>%
      select(input$check_cols)
  )
  output$data_cols_info <- renderUI({ HTML(info$column_info[is.element(info$column_name, input$check_cols)]) })
  # Plots
  ## Bar
  output$barplot <- renderPlot(
    ggplot(group_by(if (input$barplot_subject == "Math") { mat } else { port }, input$barplot_x),
           aes_string(x = input$barplot_x, fill = paste("factor(", input$barplot_fill, ")",sep = ""))) +
      geom_bar(color = "black") +
      facet_wrap(if (input$barplot_split_switch == "TRUE") { reformulate(input$barplot_split) } else { NULL }) +
      ggtitle(paste(input$barplot_subject, 'Course'), if (input$barplot_split_switch == "TRUE") { input$barplot_split } else { "" }) +
      xlab(input$barplot_x) +
      ylab(input$barplot_y) +
      labs(fill = input$barplot_fill) +
      theme_bw() +
      theme(plot.title = element_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5))
  )
  output$barplot_x_info <- renderUI({ HTML(info$column_info[info$column_name == input$barplot_x]) })
  output$barplot_fill_info <- renderUI({ HTML(info$column_info[info$column_name == input$barplot_fill]) })
  output$barplot_split_info <- renderUI({ HTML(info$column_info[info$column_name == input$barplot_split]) })
```

Serwer cd.

```
    prediction_parameters <- reactive({ data.table(school = input$pred_school,
                                                   sex = input$pred_sex,
                                                   age = as.numeric(input$pred_age),
                                                   address = input$pred_address,
                                                   famsize = input$pred_famsize,
                                                   Pstatus = input$pred_Pstatus,
                                                   Medu = input$pred_Medu,
                                                   Fedu = input$pred_Fedu,
                                                   Mjob = input$pred_Mjob,
                                                   Fjob = input$pred_Fjob,
                                                   reason = input$pred_reason,
                                                   guardian = input$pred_guardian,
                                                   traveltime = input$pred_traveltime,
                                                   studytime = input$pred_studytime,
                                                   failures = input$pred_failures,
                                                   schoolsup = input$pred_schoolsup,
                                                   famsup = input$pred_famsup,
                                                   paid = input$pred_paid,
                                                   activities = input$pred_activities,
                                                   nursery = input$pred_nursery,
                                                   higher = input$pred_higher,
                                                   internet = input$pred_internet,
                                                   romantic = input$pred_romantic,
                                                   famrel = input$pred_famrel,
                                                   freetime = input$pred_freetime,
                                                   goout = input$pred_goout,
                                                   Dalc = input$pred_Dalc,
                                                   Walc = input$pred_Walc,
                                                   health = input$pred_health,
                                                   absences = as.numeric(input$pred_absences)) })
    output$pred_param1 <- renderTable({ prediction_parameters()[,1:10] })
    output$pred_param2 <- renderTable({ prediction_parameters()[,11:20] })
    output$pred_param3 <- renderTable({ prediction_parameters()[,21:30] })

    prediction_output_mat <- eventReactive(input$button_pred, {
        paste("Predicted Math Grade:", round(predict(model_mat, newdata = prediction_parameters())))
    })
    prediction_output_por <- eventReactive(input$button_pred, {
        paste("Predicted Portuguese Grade:", round(predict(model_por, newdata = prediction_parameters())))
    })
    output$pred_output_mat <- renderText({
        prediction_output_mat()
    })
    output$pred_output_por <- renderText({
        prediction_output_por()
    })
```

Dziękujemy za uwagę