

Aplicación Shiny para Regresión Logística Interactiva

Tu Nombre

Análisis Estadístico con R Shiny

Código Fuente Completo

Índice

1. Introducción	2
2. Características Principales	2
3. Código Fuente	2
3.1. Librerías Requeridas	2
3.2. Datos de Ejemplo	2
3.3. Interfaz de Usuario (UI)	3
3.4. Servidor (Server)	6
3.5. Ejecución de la Aplicación	8
4. Conclusiones	8

1. Introducción

Esta aplicación Shiny proporciona una interfaz interactiva para realizar análisis de regresión logística. Permite a los usuarios cargar sus propios datos o utilizar un conjunto de datos de ejemplo sobre admisión universitaria.

2. Características Principales

- Carga de datos desde archivo CSV o uso de datos de ejemplo
- Selección dinámica de variables predictoras y respuesta
- Visualización interactiva de los datos
- Ajuste automático del modelo de regresión logística
- Matriz de confusión y métricas de desempeño
- Sección educativa con conceptos teóricos

3. Código Fuente

3.1. Librerías Requeridas

```
1 library(shiny)
2 library(ggplot2)
3 library(DT)
4 library(bslib)
```

3.2. Datos de Ejemplo

```
1 # Datos de ejemplo: admision universitaria
2 datos_ejemplo <- data.frame(
3   gre = c(380, 660, 800, 640, 520, 760, 560, 400, 540, 700,
4           800, 440, 760, 700, 700, 480, 780, 580, 500, 660),
5   gpa = c(3.61, 3.67, 4.00, 3.19, 2.93, 3.00, 2.98, 3.08, 3.39, 3.92,
6           4.00, 3.22, 3.08, 2.84, 3.28, 3.44, 3.87, 3.76, 2.42, 3.52),
7   admitido = c(0, 1, 1, 1, 0, 1, 0, 0, 0, 1,
8                1, 0, 1, 1, 0, 0, 1, 1, 0, 1)
9 )
```

3.3. Interfaz de Usuario (UI)

```

1 ui <- page_sidebar(
2   title = "Regresion Logistica - Analisis Interactivo",
3   theme = bs_theme(version = 5, bootswatch = "flatly"),
4
5   sidebar = sidebar(
6     width = 300,
7     h4("Configuracion"),
8
9     radioButtons("data_source", "Fuente de datos:",
10                  choices = c("Datos de ejemplo" = "ejemplo",
11                              "Subir archivo CSV" = "subir")),
12
13     conditionalPanel(
14       condition = "input.data_source == 'subir'",
15       fileInput("file", "Seleccionar archivo CSV:",
16                 accept = c(".csv"))
17     ),
18
19     conditionalPanel(
20       condition = "input.data_source == 'ejemplo' || output.file_
21       uploaded",
22       uiOutput("var_y"),
23       uiOutput("var_x1"),
24       uiOutput("var_x2"),
25       actionButton("ejecutar", "Ejecutar Modelo",
26                     class = "btn-primary w-100 mt-3")
27     ),
28
29   navset_card_tab(
30     nav_panel("Datos",
31               DTOutput("tabla_datos")
32             ),
33
34     nav_panel("Visualizacion",
35               plotOutput("plot_datos", height = "500px")
36             ),
37
38     nav_panel("Resultados",
39               h4("Resumen del Modelo"),
40               verbatimTextOutput("summary_modelo"),
41               hr(),
42               h4("Matriz de Confusion"),
43               verbatimTextOutput("confusion_matrix"),
44               hr(),
45               h4("Metricas de Desempeno"),
46               uiOutput("metricas")
47             ),
48
49     nav_panel("Conceptos",
50               h3("Que es la Regresion Logistica?"),
51               p("La regresion logistica es un modelo estadistico
52               utilizado
53               para predecir la probabilidad de que una observacion

```

```

53     pertenezca a una de dos categorias (variable binaria).")
54     ,
55     h4("Funcion Logistica"),
56     p("La regresion logistica utiliza la funcion logistica
57       (sigmoide):"),
58     withMathJax("$$$P(Y=1|X) = \frac{1}{1 + e^{-((\beta_0 +
59       \beta_1X_1 + \beta_2X_2 + ... + \beta_nX_n)
60     }}}$"),
61     h4("Caracteristicas Principales"),
62     tags$ul(
63       tags$li(strong("Variable respuesta:"),
64         " Binaria (0 o 1, Si o No, Exito o Fracaso)"),
65       tags$li(strong("Variables predictoras:"),
66         " Pueden ser continuas o categoricas"),
67       tags$li(strong("Interpretacion:"),
68         " Los coeficientes representan el cambio en el
69     log-odds"),
70       tags$li(strong("Odds Ratio:"),
71         " exp(beta) indica el cambio multiplicativo en
72     las
73         probabilidades")
74     ),
75     h4("Supuestos del Modelo"),
76     tags$ol(
77       tags$li("La variable dependiente es binaria"),
78       tags$li("Las observaciones son independientes"),
79       tags$li("Poca o ninguna multicolinealidad entre
80     predictores"),
81       tags$li("Linealidad entre variables independientes y log
82     -odds"),
83       tags$li("Muestra grande (tipicamente n > 50)")
84     ),
85     h4("Metricas de Evaluacion"),
86     tags$ul(
87       tags$li(strong("Accuracy (Exactitud):"),
88         " Proporcion de predicciones correctas"),
89       tags$li(strong("Sensibilidad (Recall):"),
90         " Proporcion de positivos correctamente
91     identificados"),
92       tags$li(strong("Especificidad:"),
93         " Proporcion de negativos correctamente
94     identificados"),
95       tags$li(strong("Precision:"),
96         " Proporcion de predicciones positivas que son
97     correctas")
98     ),
99     h4("Aplicaciones Comunes"),
100    tags$ul(
101      tags$li("Prediccion de enfermedades (diagnostico medico)
102    "),
103      tags$li("Aprobacion de creditos bancarios"),

```

```
99         tags$li("Prediccion de abandono de clientes (churn)"),
100         tags$li("Admision universitaria"),
101         tags$li("Clasificacion de spam en emails")
102     )
103 )
104 )
105 )
```

3.4. Servidor (Server)

```
1 server <- function(input, output, session) {
2
3   datos <- reactive({
4     if (input$data_source == "ejemplo") {
5       return(datos_ejemplo)
6     } else {
7       req(input$file)
8       tryCatch({
9         read.csv(input$file$datapath)
10      }, error = function(e) {
11        showNotification("Error al cargar el archivo", type = "error")
12        return(NULL)
13      })
14    }
15  })
16
17  output$file_uploaded <- reactive({
18    !is.null(input$file)
19  })
20  outputOptions(output, "file_uploaded", suspendWhenHidden = FALSE)
21
22  output$var_y <- renderUI({
23    req(datos())
24    selectInput("y_var", "Variable respuesta (binaria):",
25              choices = names(datos()))
26  })
27
28  output$var_x1 <- renderUI({
29    req(datos())
30    selectInput("x1_var", "Variable predictora 1:",
31              choices = names(datos()))
32  })
33
34  output$var_x2 <- renderUI({
35    req(datos())
36    selectInput("x2_var", "Variable predictora 2:",
37              choices = names(datos()))
38  })
39
40  output$tabla_datos <- renderDT({
41    req(datos())
42    datatable(datos(), options = list(pageLength = 10))
43  })
44
45  modelo <- eventReactive(input$ejecutar, {
46    req(datos(), input$y_var, input$x1_var, input$x2_var)
47
48    df <- datos()
49    formula_str <- paste(input$y_var, "~", input$x1_var, "+", input$x2_var)
50
51    tryCatch({
52      glm(as.formula(formula_str), data = df, family = binomial())
53    }, error = function(e) {
```

```

54     showNotification(paste("Error en el modelo:", e$message), type = "
error")
55     return(NULL)
56   })
57 })
58
59 output$plot_datos <- renderPlot({
60   req(datos(), input$y_var, input$x1_var, input$x2_var)
61
62   df <- datos()
63   df$y_factor <- factor(df[[input$y_var]])
64
65   ggplot(df, aes_string(x = input$x1_var, y = input$x2_var,
66                         color = "y_factor")) +
67     geom_point(size = 4, alpha = 0.7) +
68     scale_color_manual(values = c("0" = "#e74c3c", "1" = "#2ecc71"),
69                        labels = c("0" = "No", "1" = "Si")) +
70     labs(title = "Distribucion de Datos por Clase",
71          x = input$x1_var,
72          y = input$x2_var,
73          color = input$y_var) +
74     theme_minimal(base_size = 14) +
75     theme(legend.position = "top")
76 })
77
78 output$summary_modelo <- renderPrint({
79   req(modelo())
80   summary(modelo())
81 })
82
83 output$confusion_matrix <- renderPrint({
84   req(modelo(), datos())
85
86   df <- datos()
87   predicciones <- ifelse(predict(modelo(), type = "response") > 0.5,
88                           1, 0)
89   actual <- df[[input$y_var]]
90
91   cat("Matriz de Confusion:\n\n")
92   print(table(Predicho = predicciones, Real = actual))
93 })
94
95 output$metricas <- renderUI({
96   req(modelo(), datos())
97
98   df <- datos()
99   predicciones <- ifelse(predict(modelo(), type = "response") > 0.5,
100                           1, 0)
101   actual <- df[[input$y_var]]
102
103   cm <- table(predicciones, actual)
104
105   accuracy <- sum(diag(cm)) / sum(cm)
106   sensitivity <- cm[2, 2] / sum(cm[, 2])
107   specificity <- cm[1, 1] / sum(cm[, 1])
108   precision <- cm[2, 2] / sum(cm[2, ])

```



```
107 tagList(  
108   div(class = "row",  
109     div(class = "col-md-3",  
110       div(class = "card bg-primary text-white mb-3",  
111         div(class = "card-body",  
112           h5("Exactitud"),  
113           h3(sprintf("%.2f%%", accuracy * 100))  
114         )  
115       )  
116     ),  
117     div(class = "col-md-3",  
118       div(class = "card bg-success text-white mb-3",  
119         div(class = "card-body",  
120           h5("Sensibilidad"),  
121           h3(sprintf("%.2f%%", sensitivity * 100))  
122         )  
123       )  
124     ),  
125     div(class = "col-md-3",  
126       div(class = "card bg-info text-white mb-3",  
127         div(class = "card-body",  
128           h5("Especificidad"),  
129           h3(sprintf("%.2f%%", specificity * 100))  
130         )  
131       )  
132     ),  
133     div(class = "col-md-3",  
134       div(class = "card bg-warning text-white mb-3",  
135         div(class = "card-body",  
136           h5("Precision"),  
137           h3(sprintf("%.2f%%", precision * 100))  
138         )  
139       )  
140     )  
141   )  
142 )  
143 }  
144 })  
145 }
```

3.5. Ejecución de la Aplicación

```
1 shinyApp(ui, server)
```

4. Conclusiones

Esta aplicación Shiny proporciona una herramienta educativa e interactiva para comprender y aplicar modelos de regresión logística. Los usuarios pueden experimentar con diferentes variables y visualizar los resultados de forma inmediata.