Anexo

Aplicación web con RShiny

```
1  library(readxl)
2  library(shiny)
3
4  ui <- fluidPage(
5      titlePanel("Modelo de clasificacion mediante una regresion
           logistica"),
6      tags$style(".descripcion { color: black; font-size: 20px; }")
           ,
7      h2(class = "descripcion", "Mediante esta aplicacion podemos
           clasificar un conjunto de datos a partir de un modelo de
           regresion logistica, donde tienes que introducir ciertos
           parametros para obtener una clasificacion deseada."),
8      fileInput("datos", "Seleccione la base de datos de la que
           quiere obtener resultados:", accept = ".xlsx"),
9      numericInput("seed", "Elija la semilla que crea mas
           conveniente:", value = 12),
10     numericInput("filas_de_entrenamiento", "Seleccione el numero
           de filas destinadas al entrenamiento:", value = NULL),
11     numericInput("filas_de_testeo", "Seleccione el numero de
           filas destinadas al testeo:", value = NULL),
12     h2(class = "descripcion", "Los resultados de clasificacion
           obtenidos son:"),
13     verbatimTextOutput("filas_de_prediccion"),
14     verbatimTextOutput("bondad_ajuste"),
15     tableOutput("datos_prediccion")
16 )
17 server <- function(input, output, session) {
18     output$filas_de_prediccion <- renderText({
19       datos <- data.frame(read_excel((input$datos)$datapath,1))
20
21       filas_de_prediccion  <- nrow(datos)-(input$filas_de_
           entrenamiento)-(input$filas_de_testeo)
22
23       paste("El numero de filas destinadas para la base de
           prediccion es:", filas_de_prediccion)
24     }
25
```

```
26      )
27      output$bondad_ajuste <- renderText({
28
29        datos <- data.frame(read_excel((input$datos)$datapath,1))
30        columnas<-ncol(datos)
31        datos[,1]<-factor(datos[,1])
32
33        invisible(lapply((1:columnas)[unlist(lapply(1:columnas,
              function(i){class(datos[,i])}))=="character"],function(i
              ){datos[,i]<<-factor(datos[,i])}))
34
35
36        set.seed(input$seed)
37
38        trainIndex<-sort(sample(1:nrow(datos),input$filas_de_
              entrenamiento))
39        testIndex<-sort(sample(setdiff(1:nrow(datos),trainIndex),
              input$filas_de_testeo))
40        predictionIndex<-setdiff(1:nrow(datos),c(trainIndex,
              testIndex))
41
42        datos_train <- datos[trainIndex,]
43        datos_test <- datos[testIndex,]
44        datos_prediction<-datos[predictionIndex,-1]
45
46
47        entrenamiento<-function(datos_train)
48        {
49          suppressWarnings({
50            set.seed(input$seed)
51            modelo<-glm(ESTADO ~., data = datos_train, family =
                  binomial(link = "logit"))
52          })
53          return(modelo)
54        }
55
56        testeo<-function(datos_test,modelo)
57        {
58          suppressWarnings({
59            set.seed(input$seed)
60            bondaddelajuste<-100*sum(unlist(lapply(predict(modelo,
                  datos_test[,2:columnas],type="response"),function(i)
                  {if(i<=0.5){return(0)}else{return(1)}}))==datos_test
                  [,1])/nrow(datos_test)
61          })
62          return(bondaddelajuste)
63        }
64
65
```

```
66          modelo<-entrenamiento(datos_train)
67          bondaddelajuste<-testeo(datos_test,modelo)
68
69          paste("La bondad del ajuste obtenida es:", bondaddelajuste,
                "%")
70
71        })
72
73
74
75        output$datos_prediccion<- renderTable({
76
77          datos <- data.frame(read_excel((input$datos)$datapath,1))
78          columnas<-ncol(datos)
79          datos[,1]<-factor(datos[,1])
80
81          invisible(lapply((1:columnas)[unlist(lapply(1:columnas,
                function(i){class(datos[,i])}))=="character"],function(i
                ){datos[,i]<<-factor(datos[,i])}))
82
83
84          set.seed(input$seed)
85
86          trainIndex<-sort(sample(1:nrow(datos),input$filas_de_
                entrenamiento))
87          testIndex<-sort(sample(setdiff(1:nrow(datos),trainIndex),
                input$filas_de_testeo))
88          predictionIndex<-setdiff(1:nrow(datos),c(trainIndex,
                testIndex))
89
90          datos_train <- datos[trainIndex,]
91          datos_test <- datos[testIndex,]
92          datos_prediction<-datos[predictionIndex,-1]
93
94
95          entrenamiento<-function(datos_train)
96          {
97            suppressWarnings({
98              set.seed(input$seed)
99              modelo<-glm(ESTADO ~., data = datos_train, family =
                    binomial(link = "logit"))
100           })
101           return(modelo)
102         }
103
104         prediccion<-function(datos_prediction,modelo)
105         {
106           suppressWarnings({
107             set.seed(input$seed)
```

```
108             prediccion<-unlist(lapply(predict(modelo,datos_
                    prediction,type="response"),function(i){if(i<=0.5){
                    return(0)}else{return(1)}}))
109           })
110           return(cbind("RESPUESTA"=prediccion,datos_prediction))
111         }
112
113         modelo<-entrenamiento(datos_train)
114         prediccion<-prediccion(datos_prediction,modelo)
115
116         prediccion
117       })
118 }
119
120 shinyApp(ui = ui, server = server)
```

Código en RStudio

```
1  library(readxl)
2
3  datos <- data.frame(read_excel("datos.xlsx"))
4  columnas<-ncol(datos)
5  datos[,1]<-factor(datos[,1])
6
7  invisible(lapply((1:columnas)[unlist(lapply(1:columnas,function(i
       ){class(datos[,i])}))=="character"],function(i){datos[,i]<<-
       factor(datos[,i])}))
8
9  seed<-3
10 set.seed(seed)
11
12 trainIndex<-sort(sample(1:nrow(datos),2360))
13 testIndex<-sort(sample(setdiff(1:nrow(datos),trainIndex),2360))
14 predictionIndex<-setdiff(1:nrow(datos),c(trainIndex,testIndex))
15
16 datos_train <- datos[trainIndex,]
17 datos_test <- datos[testIndex,]
18 datos_prediction<-datos[predictionIndex,-1]
19
20
21 nrow(datos)-nrow(datos_train)-nrow(datos_test)
22
23 entrenamiento<-function(datos_train)
24 {
25   suppressWarnings({
26   set.seed(seed)
27   modelo<-glm(ESTADO ~., data = datos_train, family = binomial(
         link = "logit"))
28   })
```

```
29    return(modelo)
30  }
31
32  testeo<-function(datos_test,modelo)
33  {
34    suppressWarnings({
35    set.seed(seed)
36    bondaddelajuste<-100*sum(unlist(lapply(predict(modelo,datos_
          test[,2:columnas],type="response"),function(i){if(i<=0.5){
          return(0)}else{return(1)}}))==datos_test[,1])/nrow(datos_
          test)
37    })
38    return(bondaddelajuste)
39  }
40
41  prediccion<-function(datos_prediction,modelo)
42  {
43    suppressWarnings({
44    set.seed(seed)
45    prediccion<-unlist(lapply(predict(modelo,datos_prediction,type=
          "response"),function(i){if(i<=0.5){return(0)}else{return(1)
          }}))
46    })
47    return(cbind("RESPUESTA"=prediccion,datos_prediction))
48  }
49
50  modelo<-entrenamiento(datos_train)
51  bondaddelajuste<-testeo(datos_test,modelo)
52  prediccion<-prediccion(datos_prediction,modelo)
```