Developing Data Products Course Project

Executive Summary

This simple shiny app uses two regression models, simple linear regression model and readom forest, to predict the duration of eruption time based on the observed Old Faithful Geyser Data which contains 272 observations of the waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park.

Before Using the App

Before using the shiny app, you need to have the following packages and libraries installed

```
library(shiny)
library(devtools)
library(shinyapps)
library(caret)
library(randomForest)
```

The shiny app is deplyed to Shinyapps.io site http://georgeli88.shinyapps.io/oldfaithful.

If you want to try locally, please download **server.R** and **ui.R** into the same local folder and then run command **runApp()**.

How to Use the App

1. Enter a positive number for the waiting time
2. Select the regression model, "lm" or "randomForest"
3. Click "Submit" button
4. The predicted eruption time is displayed in the main panel
5. If an non-positive number is entered, a warning message is displayed in the main panel

Appendix

ui.R

```
library(shiny)

shinyUI(pageWithSidebar(
        headerPanel("Old Faithful Eruption Time Prediction Using Regression Model"),
        sidebarPanel(
                numericInput("inputTime", "Enter a positive waiting time (in mins)", value =50),
                radioButtons("radio", "Select Regression Model",
                             choices = list("Use lm" = 1, "Use randomForest" = 2),selected = 1),
```

```
                actionButton("goButton", "Submit")
        ),
        mainPanel(
                #h3('Predicted eruption time (in mins)'),
                h4('Based on the observed Old Faithful Geyser Data which contains 272 observations of
                   the waiting time between eruptions and the duration of the eruption for the Old
                   Faithful geyser in Yellowstone National Park, enter a waiting time between eruptions
                   to pedict the duration of eruption time.'),
                h4('Waiting time you entered (in mins):'),
                verbatimTextOutput("inputValue"),
                h4('Eruption time predicted (in mins):'),
                verbatimTextOutput("prediction")
        )
))
```

server.R

```
library(shiny)
library(caret)
library(randomForest)

# the following code gets called once
data(faithful)

# build a simple linear regression model
set.seed(12345)
inTrain <- createDataPartition(y=faithful$waiting, p=0.70, list=FALSE)
trainFaith <- faithful[inTrain, ]
testFaith <- faithful[-inTrain, ]
model1 <- lm(eruptions ~ waiting, data=trainFaith)
model2 <- randomForest(eruptions ~ waiting, data=trainFaith)

shinyServer(
        function(input, output) {
                output$inputValue <- renderText({

                        if (input$goButton==0)
                                isolate("")
                        else
                                isolate(input$inputTime)

                })

                output$prediction <- renderText({

                        if (input$goButton==0)
                                isolate("")
                        else
                        {
                                isolate({
                                t <- as.integer(input$inputTime)
                                newt <- data.frame(waiting=t)
```

```
                                if (t<=0)
                                        isolate("Warning: re-enter a number > 0")
                                else
                                        # do prediction with the entered number
                                        if (input$radio ==1 )
                                                predict(model1, newt)
                                        else
                                                predict(model2, newt)
                                })
                        }
                })

        }
)
```

User Interface