МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Кубанский государственный университет»**

**(ФГБОУ ВО «КубГУ»)**

Факультет математики и компьютерных наук

Кафедра математических и компьютерных методов

**Дневник**

**учебной практики  
(научно-исследовательская работа**

**(получение первичных навыков научно-исследовательской работы))**

студента Пасько Дмитрия Анатольевича группа 102/1

Направление подготовки 02.04.01 Математика и компьютерные науки

Программа магистратуры Математическое и компьютерное моделирование

Форма обучения очная

Квалификация магистр

Краснодар 2020

Срок прохождения практики: с 30.05.2020 по 14.06.2020

|  |  |  |  |
| --- | --- | --- | --- |
| Срок  выполнения | Краткое содержание работы практиканта | Отметка о  выполнении работы | Подпись  руководителя  практики |
| 30.05.2020-2.06.2020 | Сбор данных о средствах разработки математических веб-приложений |  |  |
| 3.06.2020-7.06.2020 | Создание пользовательской части приложения и функций-обёрток |  |  |
| 8.06.2020-11.06.2020 | Создание серверной части приложения |  |  |
| 12.06.2020-13.06.2020 | Публикация приложения, тестирование, исправление ошибок |  |  |
| 14.06.2020 | Оформление результатов |  |  |

Студент-практикант \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пасько Д. А.

СОДЕРЖАНИЕ

Введение

Описание основных компонент программы

Результаты

Заключение

# ВВЕДЕНИЕ

В эпоху цифровых технологий неоспоримой является необходимость легкого и, возможно даже, интерактивного доступа к результатам научных исследований. Мало пользы имеет такое исследование, которое просто где-то опубликуют, фактически оставив неизвестным для множества людей, заинтересованных в его результатах. От первой официально опубликованной мной статьи остался лишь номер со ссылкой на какой-то печатный сборник с очень ограниченным тиражом, так что даже я, автор, не имею возможности её найти и прочесть.

Особенно верно сказанное выше для научных работ, нацеленных принести пользу группам людей, далёким от конкретных наук (физики, биологии, математики, экономики, социологии). Таким людям очень трудно найти действительно понятные и полезные статьи по интересующему вопросу, не то что извлечь оттуда применимую на практике информацию. Кроме того, имеют место разные ментальные ограничения: многие люди просто не верят в способность науки решить некоторые их проблемы. В моём случае была поставлена цель создать регрессионную модель для определения повторного максимума в силовых упражнениях. Но мало создать такую модель – в чём её смысл, если люди не знают о её существовании или не понимают, как ей пользоваться?..

Описанная модель распространения научного знания несовершенна и постепенно будет уходить в прошлое. Обычные научные статьи проиграют в конкурентной борьбе более совершенным методам, использующим преимущества современных технологий. В ближайшем будущем уже никто не захочет читать статьи с не совсем понятными графиками из Excel или Statistica: конечно, если очень нужно, мы можем закрыть глаза на плохое качество графиков и устаревшие паттерны расстановки объектов на них, но вряд ли мы станем прощать подобные недостатки систематически, зная, что уже десяток лет существуют прекрасные графические библиотеки matplotlib (Python) или ggplot2 (R): статьи старого формата просто не будут читаться, как сегодня не читаются старые, пусть и качественные в плане содержания, советские издания по математике, использующие уродливые формулы, а не формулы формата LaTeX.

Точно так же едва ли вызовут интерес работы по, например, распознаванию речи или переводу, если к ним не будет приложен пакет PyPi или Telegram bot.

Точно так же передо мной встала задача сделать результаты исследования доступнее «целевой аудитории». Речь идёт о создании веб-приложения, в котором спортсмен вводит свои (или любые другие) параметры и получает предсказания своего повторного максимума.

# Описание основных компонент программы

Поскольку сама регрессионная модель написана на языке R и является объектом R, её очень легко использовать в приложениях R Shiny.

Сперва подключим нужные библиотеки (сам фреймворк и библиотеки для функционального программирования):

library(shiny)

library(magrittr)

library(dplyr)

Опишем компоненты, видимые пользователем (user interface):

ui <- fluidPage(

# Application title

titlePanel("Estimating repeated maximim by multi-repeated maximum"),

# Sidebar with a slider input for number of bins

sidebarLayout(

sidebarPanel(

numericInput(inputId = "MRM",

label = "Your repeated maximum(kg):",

value = 100,min=20,max=450,

step =2.5,

width = "30%"),

selectInput(inputId = "Count",

label = "Repeats:",

choices = 2:10 %>% as.character(),

width = "30%"),

selectInput(inputId = "Action",

label = "Choose an action:",

choices = c("Bench press", "Squat", "Deadlift"),

width = "40%"),

sliderInput("Weight",

"Choose your weight(kg):",

min = 40,

max = 160,

value = 80),

sliderInput("Height",

"Choose your height(cm):",

min = 150,

max = 210,

value = 180)

),

# Show a plot of the generated distribution

mainPanel(

h3(textOutput("cap1", container = span)),

tableOutput("vals"),

h3(textOutput("cap2", container = span)),

tableOutput("vals2"),

textOutput("sorry"),

h3(textOutput("cap3", container = span)),

tableOutput("perc")

)

)

)

Суть происходящего – добавление на чистую панель разных объектов (надписи, слайдеры) с конкретными параметрами и именами для обращения к ним с серверной части. Но сперва требуется загрузить регрессионную модель и написать несколько функций-обёрток для её использования:

e = new.env()

name <- load("entire\_data.rdata", envir = e)

f=function(MRM,Count,Action='Жим',Weight=70,Height=170){

act=factor(Action,levels =e$action.levels)

up=c(4,8,11)

lv=e$count.levels

cg=lv[Count<up] %>% first() %>% factor(levels=lv)

df=data.frame(MRM=MRM,

Count=Count,

Action=act,

CountGroup=cg,

Index=Weight/(0.01\*Height)^2)

predict(e$b5,

df,

se.fit = T,

interval = "confidence",

level=0.999)[[1]] %>% return()

}

f2=function(MRM,Count,Action='Жим',Weight=70,Height=170){

act=factor(Action,levels =e$action.levels)

up=c(4,8,11)

lv=e$count.levels

cg=lv[Count<up] %>% first() %>% factor(levels=lv)

df=data.frame(MRM=MRM,

Count=Count,

Action=act,

CountGroup=cg,

Index=Weight/(0.01\*Height)^2

predict(e$n14, df) %>% return()

}

mrm3=function(RM,count,Action='Присед',Weight=70,Height=170){

cf=e$cf

ctg=3

if(count<7){ctg=2}

if(count<4){ctg=1}

act=0

if(Action=="Тяга"){

act=cf[5]}else if(Action=="Присед"){act=cf[6]}

polyroot(c(-RM,cf[1+ctg]+count\*cf[6+ctg]+act,0,0,0,0,cf[1]\*((0.01\*Height)^2/Weight)^6))[1] %>% Re()

}

Серверная часть приложения (описанным ранее объектам даётся поведение):

server <- function(input, output) {

output$cap1 <- renderText({

'Predictions by linear model:'

})

output$cap2 <- renderText({

'Predicted multi-reps:'

})

output$cap3 <- renderText({

'Percentiles:'

})

output$sorry<-renderText({

'The present model can give outliers if RM or Repeats are very large or the body mass index is too small. It`s because of features of the dataset, sorry'

})

output$vals <- renderTable({

v=f(input$MRM,

input$Count %>% as.numeric(),

Action=switch (input$Action,

"Bench press" = "Жим",

"Squat" = "Присед",

"Deadlift" = "Тяга",

),

Height=input$Height,

Weight=input$Weight) %>% round(1)

colnames(v)=c("RM Prediction","Lower prediction","Upper prediction")

tbl\_df(v)

})

output$vals2 <- renderTable({

MRM=input$MRM

Count=input$Count %>% as.numeric()

Action=switch (input$Action,

"Bench press" = "Жим",

"Squat" = "Присед",

"Deadlift" = "Тяга",

)

Height=input$Height

Weight=input$Weight

v=f(MRM,

Count,

Action=Action,

Height=Height,

Weight=Weight)[1] %>% round(1)

v2=sapply(2:10, function(xx) mrm3(v,xx,Action = Action,Weight = Weight,Height = Height))

v2=c(v,v2 %>% round(1))

nas=v2>v

v2[nas]=NA

if(length(nas)>0){

for(i in 2:9){

if(is.na(v2[i])){

v2[i]=(v2[i-1]+v2[i+1])/2

}

}

}

inds=nas&!is.na(v2)

v2[inds]=paste('~',v2[inds])

df=data.frame(1:10,v2)

names(df)=c("Count of repeats","Predicted weight")

df

})

output$perc <- renderTable({

v=f(input$MRM,

input$Count %>% as.numeric(),

Action=switch (input$Action,

"Bench press" = "Жим",

"Squat" = "Присед",

"Deadlift" = "Тяга",

),

Height=input$Height,

Weight=input$Weight)[1]

v2=f2(input$MRM,

input$Count %>% as.numeric(),

Action=switch (input$Action,

"Bench press" = "Жим",

"Squat" = "Присед",

"Deadlift" = "Тяга",

),

Height=input$Height,

Weight=input$Weight)

vc=seq(100,50,by=-5)

g=tibble('Percentage of RM'=paste0(vc,'%'),'Linear model'=round(v\*vc/100,1),'Nonlinear model'=round(v2\*vc/100,1))

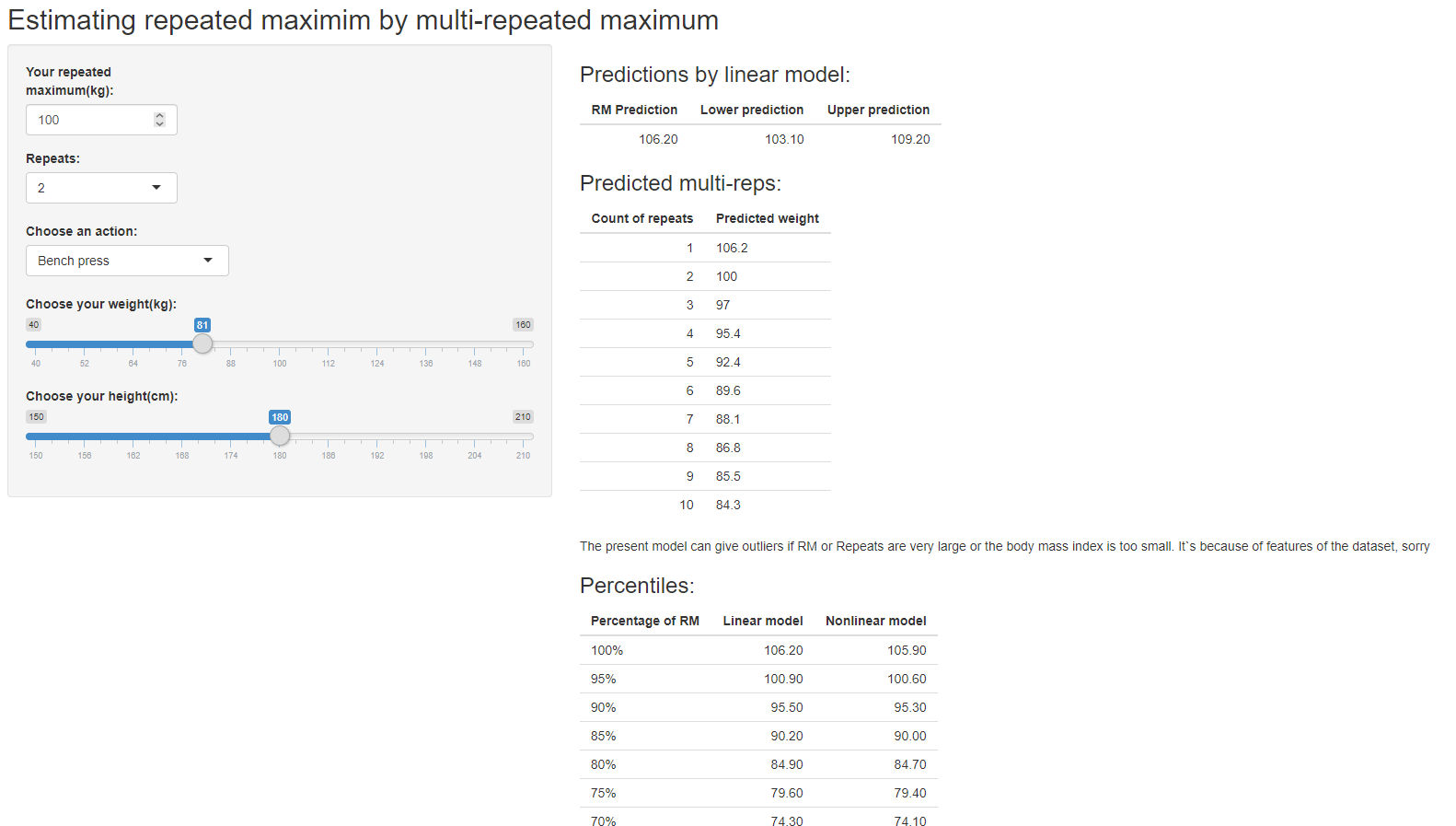
g

})

}

# Результаты

Запустив приложение, получим такой результат:



Пользователь может вводить данные и получать не только предсказание для повторного максимума, но и доверительный интервал и перцентили. Приложение опубликовано и доступно по ссылке <https://dmitrypasko.shinyapps.io/RMbyMRMestimating/>. Исходный код приложения и само исследование расположены в репозитории <https://github.com/PasaOpasen/Powerlifting-training-diary-and-articles>.

# ЗАКЛЮЧЕНИЕ

В ходе практики было разработано приложение, позволяющее протестировать результаты исследования любому спортсмену. По сути, это специализированный онлайн-калькулятор, чья логика опирается на научную работу, а не на неясно откуда берущиеся формулы.

Многие пользователи этого приложения отметили его точность и удобность. Если почти каждое научное исследование будет сопровождаться подобными дополнениями, научный прогресс, несомненно, ускорится.