Stroke Risk Factors

Kira Wolff

11 4 2021

Introduction

After a stroke occurs, every minute counts (Darehed et al. 2020). The sooner the affected person receives medical care, the higher the chances of a good recovery and little subsequent damages. Thus it is key to recognize a stroke occured. Besides knowledge about the visible symptoms of a stroke, knowledge about who is at high(er) risk to suffer a stroke can help channeling valuable attention resources on the these people.

Research Question

The goal of this project is to identify medical and/or lifestyle factors that are associated with stroke occurence by constructing a model that predicts stroke occurence and finding the important predictors.

```
library(tidyverse) # data handling
library(scales) # add percentages to bar plots
library(ggpubr) # arrange plots
library(caret) # modeling methods
library(randomForest) # to use random forest in caret
library(knitr) # table design
```

Method

To answer the research question, data about people with and without a stroke in their medical history will be used. The goal is to build a model that classifies the people correctly as stroke patients and people without stroke history. This model could then be applied to other people whose basic demographic, medical and lifestyle data is known. If the model classifies them as stroke patients although they did not experience one, they should be treated as high risk.

Simple comparisons by group will be executed with t-tests or χ^2-tests for continuous and categorial variables, respectively.

As common procedure for binary classification problems, a logistic regression will be conducted. Considering potentially more complex connections between the variables, further modeling approaches will be used. Firstly, a regularization term in form of Elastic Net will be added which is especially sensible in prediction contexts as it reduces overfitting. Elastic Net was choosen instead of LASSO or Ridge regularization because it enables a sort of compromise between the other methods, as both parameters can be tuned. Secondly, to consider the case that the relationship of the variables is better represented by a more complex function, support vector machines (SVMs) with a linear kernel will be applied. Finally, a classification tree using the random forest algorithm will be applied. This might be especially useful to identify the most important/impactful variables, that is, predictors.

To train the models and tune the parameters, repeated 5-fold cross-validation will be used. Additionally, the final models' performance will be tested with a subset of the data that is not part of the training, consisting of 20% of the original dataset. The best model will be identified via accuracy as performance criterium. Sensitivity and Specifity will also be considered, as well as the general complexity of the model - in case of near identical performance, simpler models will be chosen.

Finally, the best model will be evaluated regarding the predictors. If possible, a p-value of 0.05 will be considered as a significant result.

Implementation

The analysis will be conducted using *R* version 4.0.3 (R Core Team 2020) in *RStudio* version 1.3.1093 (RStudio Team 2020). Data will be handled via *tidyverse* (Wickham et al. 2019), primarly with *dplyr* [dplyr2020], and plotted with *ggplot2* (Wickham 2016), *scales* (Wickham and Seidel 2020) and *ggpubr* (Kassambara 2020). The models will be implemented using *caret* (Kuhn 2020) and *randomForest* (Liaw and Wiener 2002). For resampling, the methods offered by *caret* will be used. Data will be preprocessed by centering and scaling before building the models.

Data

```
# define classes
stroke <- read.csv("healthcare-dataset-stroke-data.csv") %>%
  mutate(id = as.factor(id),
         gender = as.factor(gender),
         age = as.numeric(age),
         hypertension = as.factor(hypertension),
         heart_disease = as.factor(heart_disease),
         ever_married = as.factor(ever_married),
         work_type = as.factor(work_type),
         Residence_type = as.factor(Residence_type),
         avg_glucose_level = as.numeric(avg_glucose_level),
         bmi = as.numeric(bmi),
         smoking_status = as.factor(smoking_status),
         stroke = as.factor(stroke)
         )
```

```
# function to ease identifying variables with missing data
any.na <- function(col){
  any(is.na(col))
}

# identify variables with missing data
apply(stroke[,1:length(names(stroke))], 2, FUN="any.na")
```

```
##               id           gender              age     hypertension
##            FALSE            FALSE            FALSE            FALSE
##    heart_disease     ever_married        work_type   Residence_type
##            FALSE            FALSE            FALSE            FALSE
## avg_glucose_level              bmi   smoking_status           stroke
##            FALSE             TRUE            FALSE            FALSE
```

```r
# imputation of missing data via regression
## create dataset without missing data
stroke.complete <- stroke %>%
  na.omit(bmi)

## build model
lm.bmi <- lm(bmi ~ gender + age + hypertension + heart_disease + ever_married +
               work_type + Residence_type + avg_glucose_level +
               smoking_status + stroke,
             data = stroke.complete)

## create dataset with only rows containing missing data
stroke.na <- stroke %>%
  filter(is.na(bmi))

## predict bmi
bmi.pred <- predict(lm.bmi, stroke.na)

## substitute missing data with predicted data
stroke.na$bmi <- bmi.pred

stroke.clean <- rbind(stroke.na, stroke.complete)
```

```r
summary(stroke.clean)
```

```
##        id          gender          age        hypertension heart_disease
##  67     :   1   Female:2994   Min.   : 0.08   0:4612       0:4834
##  77     :   1   Male  :2115   1st Qu.:25.00   1: 498       1: 276
##  84     :   1   Other :   1   Median :45.00
##  91     :   1                 Mean   :43.23
##  99     :   1                 3rd Qu.:61.00
##  121    :   1                 Max.   :82.00
##  (Other):5104
##  ever_married         work_type    Residence_type avg_glucose_level
##  No :1757     children     : 687   Rural:2514     Min.   : 55.12
##  Yes:3353     Govt_job     : 657   Urban:2596     1st Qu.: 77.25
##               Never_worked :  22                  Median : 91.89
##               Private      :2925                  Mean   :106.15
##               Self-employed: 819                  3rd Qu.:114.09
##                                                   Max.   :271.74
##
##       bmi                smoking_status stroke
##  Min.   :10.30   formerly smoked: 885   0:4861
##  1st Qu.:23.70   never smoked   :1892   1: 249
##  Median :28.30   smokes         : 789
##  Mean   :28.94   Unknown        :1544
##  3rd Qu.:32.93
##  Max.   :97.60
##
```

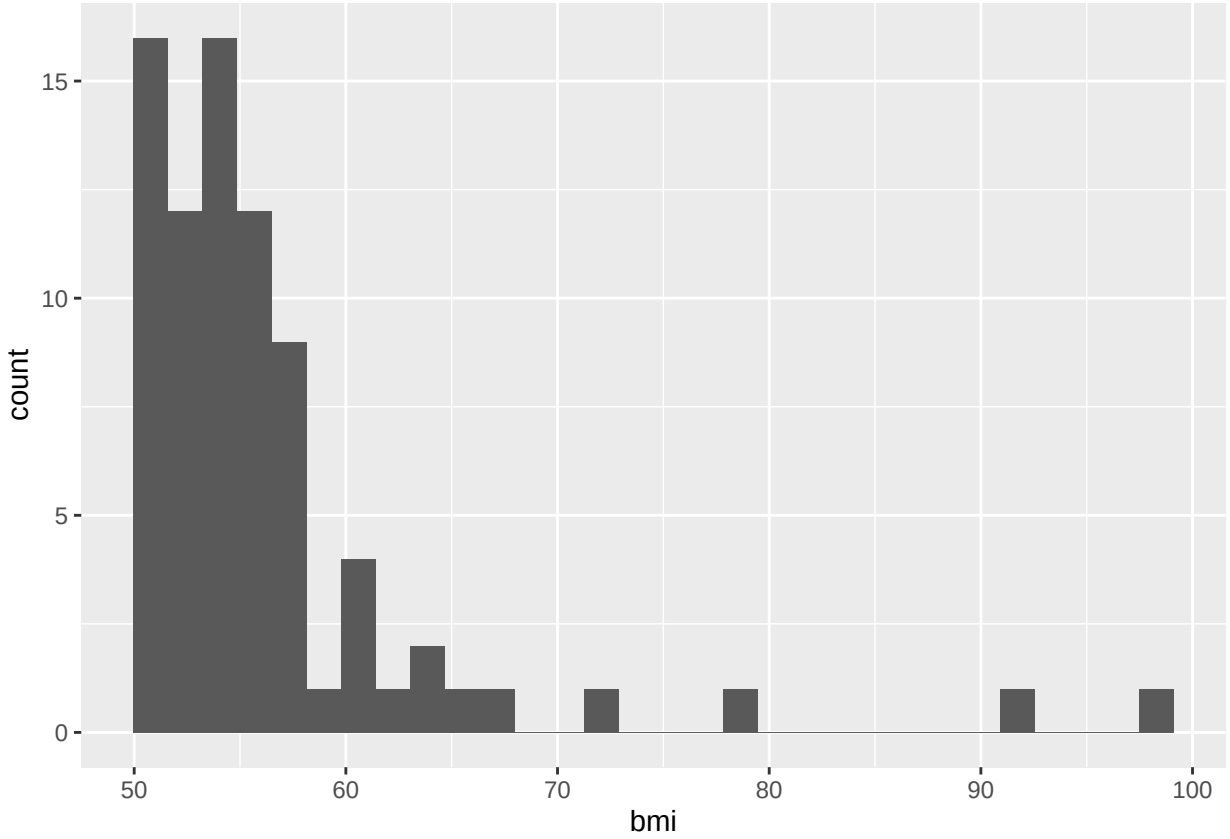
```r
# Age decimals
sum(stroke.clean$age < 1)
```

```
## [1] 43
```

```
sum(stroke.clean$age < 18)
```

```
## [1] 856
```

```
# bmi extreme
stroke.clean %>%
  filter(bmi > 50) %>%
  select(bmi) %>%
  ggplot(aes(bmi))+
    geom_histogram()
```




```
stroke.clean %>%
  filter(bmi > 70) %>%
  select(bmi)
```

```
##       bmi
## 545  71.9
## 929  78.0
## 2129 97.6
## 4210 92.0
```

```r
# smoking unknown
smok.unk <- round(sum(stroke.clean$smoking_status == "Unknown")/
                    length(stroke.clean$smoking_status), 2)*100

# outcome/stroke ratio
stroke.ratio <- stroke.clean %>%
  summarize(table(stroke)[1]/table(stroke)[2])
```

The data comes from the publicly accessible "Stroke Prediction" dataset which was published by the user fedesoriano on kaggle.net in January of 2021 (fedesoriano 2021). Unfortunately, no further information about the validity of the data is known. The dataset contains information about demographic, medical and lifestyle data from >5000 people.

The independent variable of interest, stroke occurence, is represented in the data as a binary variable, corresponding to having had a stroke in the own medical history or not. Unfortunately, the data is quite unbalanced with a ratio from 19.52:1 for the subjects not affected by a stroke.

Missing Data were handled via single imputation by predicting the missing values with a linear regression model.

A few variables show disputable values for which more context or explanation by the dataset's creator would have been useful. Since this information is not provided, I try to evaluate them as appropriate or inappropriate.

Among the age values, multiple cases show values smaller than one. Although this is not a common way to display age below one year, this is the most obvious explanation. It is not far-fetched since many other cases are underage as well. Although the age of people older than one year is just displayed as integer, the decimals might have been used for infants because an age of zero is even harder to interpret or rather incorrect. Additionally, several months make a large developmental difference for infants which could be considered here. Lastly, reducing the values to "<1" would have removed the possibility to calculate with age as a continuous variable.

Among the BMI values, outliers reach values up to 97. Although these values are extreme, they are not impossible and are thus likely valid.

Concerning the smoking status variable, information about 30% of the subjects in unknown which makes it difficult to evaluate the contribution of smoking to the research question. Nevertheless, the other extractable information from this variable can be useful.

Results

This section will describe the sample in more detail, build the models and evaluate them.

Subjects

```r
nsub <- length(stroke.clean$id)

# demographic
age.min <- min(stroke.clean$age)
age.max <- max(stroke.clean$age)
age.mean <- mean(stroke.clean$age)
age.sd <- sd(stroke.clean$age)
```

```r
fem.perc <- round(prop.table(table(stroke.clean$gender))[1]*100, 2)
men.perc <- round(prop.table(table(stroke.clean$gender))[2]*100, 2)
div.perc <- round(prop.table(table(stroke.clean$gender))[3]*100, 2)


# health
gluc.mean <- mean(stroke.clean$avg_glucose_level)
gluc.sd <- sd(stroke.clean$avg_glucose_level)

bmi.mean <- mean(stroke.clean$bmi)
bmi.sd <- sd(stroke.clean$bmi)

sick <- stroke.clean %>%
  filter(hypertension == 1 | heart_disease == 1 | stroke == 1) %>%
  summarize(n = n(),
            perc = round(n()/nsub*100, 2))


# work
employed <- stroke.clean %>%
  filter(work_type == "Govt_job" | work_type == "Private" |
           work_type == "Self-employed") %>%
  summarize(n = n(),
            perc = (n()/nsub)*100)

child <- stroke.clean %>%
  filter(work_type == "children") %>%
  summarize(n = n(),
            perc = (n()/nsub)*100)

# smoking
stroke.smoke <- stroke.clean %>%
  filter(smoking_status != "Unknown") %>%
  summarize(prop.table(table(smoking_status))*100)
```

```r
# sort variables
var.cat <- c("gender", "hypertension", "heart_disease", "ever_married",
             "work_type", "Residence_type", "smoking_status", "stroke")
var.cont <- c("age", "avg_glucose_level", "bmi")

# functions to enable "looping" plot creation
plot.bar <- function(data, x){
  ggplot(data, aes_string(x = x))+
    geom_bar(aes(y=..count../sum(..count..)),
             fill = "#4378A2")+
    ylab("Percentage")+
    scale_y_continuous(labels = percent_format())+
    theme_minimal()+
    theme(legend.title = element_blank())
}

plot.hist <- function(data, x){
  ggplot(data, aes_string(x = x))+
```

```
    geom_histogram(aes(y=..count../sum(..count..)),
                   fill = "#4378A2")+
    ylab("Percentage")+
    scale_y_continuous(labels = percent_format())+
    theme_minimal()
}

# create base plots
cat.plots <- lapply(var.cat, plot.bar, data = stroke.clean)
cont.plots <- lapply(var.cont, plot.hist, data = stroke.clean)

# refine base plots
plot.gen <- cat.plots[[1]]+
  xlab("Gender")

plot.hyp <- cat.plots[[2]]+
  xlab("Hypertension")+
  scale_x_discrete(labels = c("No", "Yes"))

plot.heart <- cat.plots[[3]]+
  xlab("Heart Disease")+
  scale_x_discrete(labels = c("No", "Yes"))

plot.mar <- cat.plots[[4]]+
  xlab("Married (Lifetime)")

plot.work <- cat.plots[[5]]+
  xlab("Work Type")+
  scale_x_discrete(labels = c("Children", "Govern. \nJob",
                              "Never \nWorked", "Private \nSector",
                              "Self-\nemployed"))

plot.res <- cat.plots[[6]]+
  xlab("Residence Type")

plot.smok <- cat.plots[[7]]+
  xlab("Smoking Status")+
  scale_x_discrete(labels = c("formerly\nsmoked", "never\nsmoked",
                              "currently\nsmokes", "unknown"))

plot.stroke <- cat.plots[[8]]+
  xlab("Stroke")+
  scale_x_discrete(labels = c("No", "Yes"))

plot.age <- cont.plots[[1]]+
  xlab("Age")+
  annotate(geom = "text",
           x = 15, y = 0.045,
           label = paste("M =",
                         round(age.mean, 2),
                         "\nSD =",
                         round(age.sd,2)))
```

```
plot.gluc <- cont.plots[[2]]+
  xlab("Glucose Level [mg/dl]")+
  annotate(geom = "text",
           x = 200, y = 0.1,
           label = paste("M =",
                         round(gluc.mean, 2),
                         "\nSD =",
                         round(gluc.sd,2)))

plot.bmi <- cont.plots[[3]]+
  xlab("BMI")+
  annotate(geom = "text",
           x = 75, y = 0.15,
           label = paste("M =",
                         round(bmi.mean, 2),
                         "\nSD =",
                         round(bmi.sd,2)))

# show prepared plots via separate chunks
```

The data represents information about 5110 subjects. The sample has a large age range from <1 to 82 years with a mean of 43.23 years ($SD = 22.61$). It consists of 58.59% women, 41.39% men, and 0.02% identifying with other labels.

```
ggarrange(plot.hyp, plot.heart, plot.stroke, plot.gluc, plot.bmi, ncol=2,
          nrow=3)
```

As Figure 1 shows, the majority of the sample has no history of hypertension, heart disease, stroke or elevated blood glucose level. Excluding the latter, 16.81% are affected by at least one of the aforementioned diagnoses overall. According to the BMI data, all categories from underweight to obese are represented by the sample (though BMI calculations for minors have to be interpreted with reservations).

```
ggarrange(plot.mar, plot.work, plot.res, plot.smok)
```

Concerning the lifestyle of the sample, the majority is married or has been at some point in their life. 86.13% were employed or self-employed at the time of data collection, while 13.44% were counted as children. Regarding their residence, the sample splitted almost fifty-fifty into rural and urban population. At last, considering only the subsample whose smoking status is known, 53.06% never smoked, while 24.82% smoked formerly and 53.06% currently still do. More information can be seen in Figure 2.

In general, the sample seems to be quite representative of the general population, without focusing on a specific subgroup. Thus, the results of the following analysis can potentially be transferred to the general population. Although one should keep in mind that some variable connections only appear in certain subgroups, so working with samples like this is not always beneficial.

```
# calculate descriptives
sample <- stroke.clean %>%
  # split by stroke
  dplyr::group_by(stroke) %>%
  # create summary
  dplyr::summarize(
    # n
```

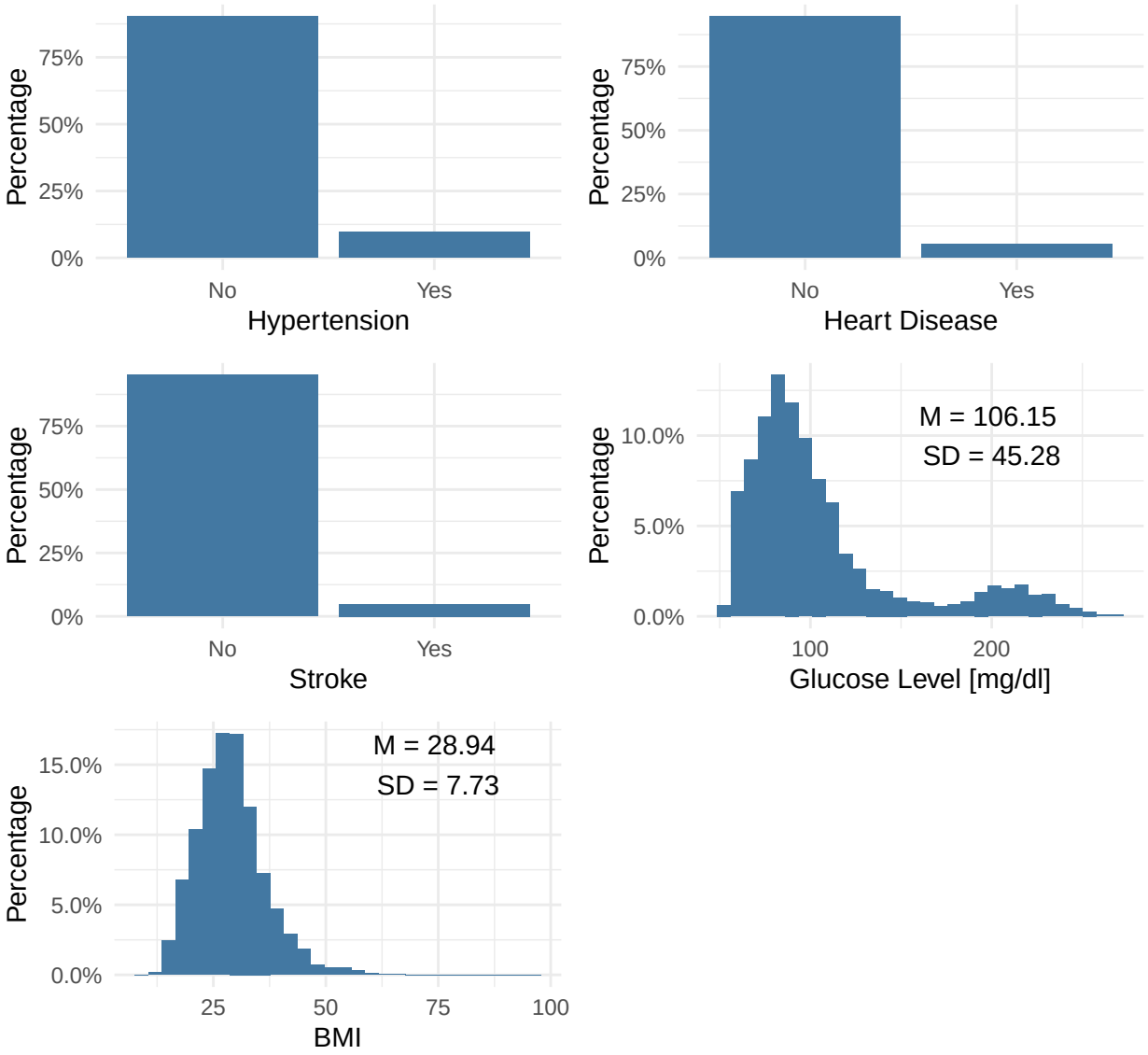


Figure 1: Distributions of Medical Data

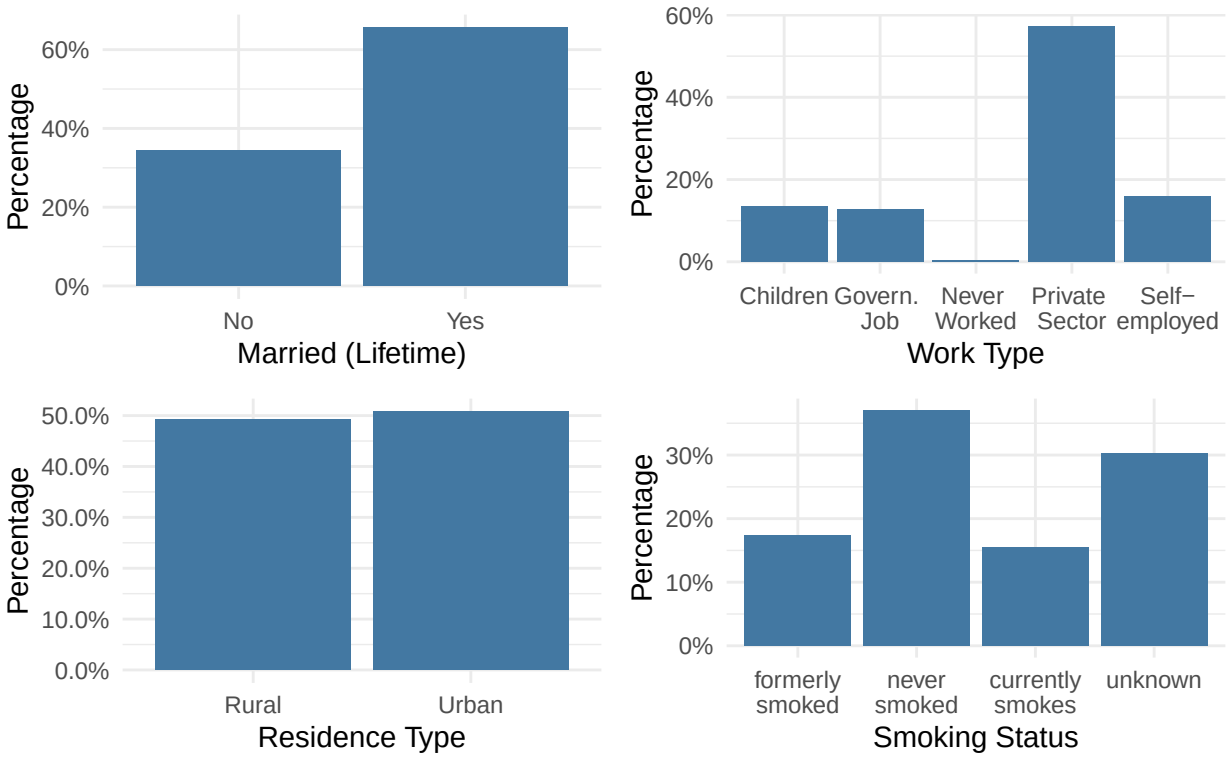


Figure 2: Distributions of Lifestyle Data

```
    n = n(),
    #gender
    'Gender (m/f)' = paste(
        round(prop.table(table(gender))[1]*100, 1),
        "/" ,
        round(prop.table(table(gender))[2]*100, 1),
        sep=""),
    #age
    Age = paste(
      round(mean(age), 1),
      " (",
      round(sd(age), 1),
      ")",
      sep=""),
    #hypertension
    'Hypertension (n/y)' = paste(
        round(prop.table(table(hypertension))[1]*100, 1),
        "/" ,
        round(prop.table(table(hypertension))[2]*100, 1),
        sep=""),
    #heart disease
    'Heart Disease (n/y)' = paste(
        round(prop.table(table(heart_disease))[1]*100, 1),
        "/" ,
        round(prop.table(table(heart_disease))[2]*100,1),
        sep=""),
```

```
    #glucose
    'Glucose [mg/dl]' = paste(
      round(mean(avg_glucose_level), 1),
      " (",
      round(sd(avg_glucose_level), 1),
      ")",
      sep=""),
    # bmi
    "BMI" = paste(
      round(mean(bmi), 1),
      " (",
      round(sd(bmi), 1),
      ")",
      sep=""),
    #married
    'Ever Married (n/y)' = paste(
        round(prop.table(table(ever_married))[1]*100, 1),
        "/" ,
        round(prop.table(table(ever_married))[2]*100, 1),
        sep=""),
    #residence
    'Residence (rural/urban)' = paste(
        round(prop.table(table(Residence_type))[1]*100, 1),
        "/" ,
        round(prop.table(table(Residence_type))[2]*100, 1),
        sep=""),
    #smoking
    'Smoking Status (formerly/never/currently)' = paste(
        round(prop.table(table(smoking_status))[1]*100, 1),
        "/" ,
        round(prop.table(table(smoking_status))[2]*100, 1),
        "/" ,
        round(prop.table(table(smoking_status))[3]*100, 1),
        sep=""),
    )

values <- t(sample[-1])

# calculate p-values

sample.sig <- stroke.clean %>%
  summarize(
    # empty for n
    placeholder = "---",
    # gender
    gender.chi = round(chisq.test(table(gender, stroke))$p.value, 3),
    # age
    age.t = round(t.test(age ~ stroke)$p.value, 3),
    # hypertension
    hyp.chi = round(chisq.test(table(hypertension, stroke))$p.value, 3),
    # heart disease
    heart.chi = round(chisq.test(table(heart_disease, stroke))$p.value, 3),
    # glucose
```

```
    gluc.t = round(t.test(avg_glucose_level ~ stroke)$p.value, 3),
    # bmi
    bmi.t = round(t.test(bmi ~ stroke)$p.value, 3),
    # married
    married.chi = round(chisq.test(table(ever_married, stroke))$p.value, 3),
    # residence
    res.chi = round(chisq.test(table(Residence_type, stroke))$p.value, 3),
    # smoking
    smoke.chi = round(chisq.test(table(smoking_status, stroke))$p.value, 3),
  )

p.values <- t(sample.sig)

# change "zero"-p-values to <.001
for(i in 1:length(p.values)){
  if(p.values[i,] == 0){
    p.values[i,] <- "<0.001"
  }
}

table1 <- cbind(values, p.values)

kable(table1,
      col.names= c("No Stroke", "Stroke", "p"),
      caption = "Sample Data by Stroke Occurence (Mean (SD) or Percentages)")
```

Table 1: Sample Data by Stroke Occurence (Mean (SD) or Percentages)

	No Stroke	Stroke	p
n	4861	249	—
Gender (m/f)	58.7/41.3	56.6/43.4	0.79
Age	42 (22.3)	67.7 (12.7)	<0.001
Hypertension (n/y)	91.1/8.9	73.5/26.5	<0.001
Heart Disease (n/y)	95.3/4.7	81.1/18.9	<0.001
Glucose [mg/dl]	104.8 (43.8)	132.5 (61.9)	<0.001
BMI	28.9 (7.8)	30.3 (5.9)	<0.001
Ever Married (n/y)	35.5/64.5	11.6/88.4	<0.001
Residence (rural/urban)	49.4/50.6	45.8/54.2	0.298
Smoking Status (formerly/never/currently)	16.8/37.1/15.4	28.1/36.1/16.9	<0.001

A comparison of the variables by stroke, as depicted in Table 1, shows the potential of the variables as predictors: For almost all variables, a significant difference emerges. Especially age and the variables concerning health, i.e. hypertension, heart disease, and glucose show large differences.

Model Evaluation

```
table(stroke.clean$stroke)
```

```
##
##    0    1
## 4861  249
```

```
summary(stroke.clean)
```

```
##        id          gender          age        hypertension heart_disease
##  67     :   1   Female:2994   Min.   : 0.08   0:4612       0:4834
##  77     :   1   Male  :2115   1st Qu.:25.00   1: 498       1: 276
##  84     :   1   Other :   1   Median :45.00
##  91     :   1                 Mean   :43.23
##  99     :   1                 3rd Qu.:61.00
##  121    :   1                 Max.   :82.00
##  (Other):5104
##  ever_married         work_type    Residence_type avg_glucose_level
##  No :1757     children     : 687   Rural:2514     Min.   : 55.12
##  Yes:3353     Govt_job     : 657   Urban:2596     1st Qu.: 77.25
##               Never_worked :  22                  Median : 91.89
##               Private      :2925                  Mean   :106.15
##               Self-employed: 819                  3rd Qu.:114.09
##                                                   Max.   :271.74
##
##       bmi                smoking_status stroke
##  Min.   :10.30   formerly smoked: 885   0:4861
##  1st Qu.:23.70   never smoked   :1892   1: 249
##  Median :28.30   smokes         : 789
##  Mean   :28.94   Unknown        :1544
##  3rd Qu.:32.93
##  Max.   :97.60
##
```

```
sum(stroke.clean$hypertension == 1 | stroke.clean$heart_disease == 1 |
      stroke.clean$stroke.clean == 1)/length(stroke.clean$id)
```

```
## [1] 0
```

```
# Split Data
# use 80% of the data to fit and tune the models, test accuracy with other 20%
train.index <- sample(1:length(stroke.clean$id),
                      size = floor(length(stroke.clean$id)*0.8))

train.data <- stroke.clean[train.index,]
test.data <- stroke.clean[-train.index,]

# Create Base Model
model <- stroke ~ gender + age + hypertension + heart_disease + ever_married +
  work_type + Residence_type + avg_glucose_level + bmi + smoking_status
```

```r
# Prepare Cross-validation
trctrl <- trainControl(method = "repeatedcv",
                       number = 5,
                       repeats = 3)
```

```r
# fit model
glm.train <- train(model,
                   train.data,
                   trControl = trctrl,
                   method = "glm",
                   preProcess = c("center", "scale"),
                   family = "binomial")

summary(glm.train)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0941  -0.3081  -0.1562  -0.0883   3.4865
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -4.051526   1.552319  -2.610 0.009055 **
## genderMale                   -0.005149   0.079915  -0.064 0.948630
## genderOther                  -0.160790  22.762828  -0.007 0.994364
## age                           1.719704   0.150170  11.452  < 2e-16 ***
## hypertension1                 0.081394   0.056713   1.435 0.151229
## heart_disease1                0.077290   0.048804   1.584 0.113267
## ever_marriedYes              -0.066068   0.124156  -0.532 0.594632
## work_typeGovt_job            -0.443143   0.291175  -1.522 0.128031
## work_typeNever_worked        -0.701655  22.602517  -0.031 0.975235
## work_typePrivate             -0.557138   0.421682  -1.321 0.186426
## `work_typeSelf-employed`     -0.594527   0.321690  -1.848 0.064583 .
## Residence_typeUrban           0.134945   0.079461   1.698 0.089461 .
## avg_glucose_level             0.208859   0.062370   3.349 0.000812 ***
## bmi                           0.014112   0.100710   0.140 0.888562
## `smoking_statusnever smoked` -0.057245   0.098086  -0.584 0.559477
## smoking_statussmokes          0.070443   0.088314   0.798 0.425079
## smoking_statusUnknown         0.014158   0.108536   0.130 0.896212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1543.2  on 4087  degrees of freedom
## Residual deviance: 1220.9  on 4071  degrees of freedom
## AIC: 1254.9
##
## Number of Fisher Scoring iterations: 14
```

```
glm.train
```

```
## Generalized Linear Model
##
## 4088 samples
##   10 predictor
##    2 classes: '0', '1'
##
## Pre-processing: centered (16), scaled (16)
## Resampling: Cross-Validated (5 fold, repeated 3 times)
## Summary of sample sizes: 3269, 3271, 3271, 3271, 3270, 3269, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9534416  0.009643826
```

```
# predict test data
glm.preds <- predict(glm.train, test.data)

# compare prediction and real data
glm.mat <- confusionMatrix(glm.preds, test.data$stroke)
```

The logistic regression provides a model with $Acc = 0.953$ which reaches $Acc = 0.943$ with the test data.

```
# fit model
reg.train <- train(model,
                   train.data,
                   method = "glmnet",
                   preProcess = c("center", "scale"),
                   metric = "Accuracy",
                   tuneLength = 10,
                   family = "binomial")

# save accuracy
reg.train.acc <- reg.train$results %>%
  filter(alpha == reg.train$bestTune$alpha &
           lambda == reg.train$bestTune$lambda) %>%
  summarize(Accuracy = Accuracy)


# predict test data
reg.preds <- predict(reg.train, test.data)

# compare prediction and real data
reg.mat <- confusionMatrix(reg.preds, test.data$stroke)
```

The hyperparameters of the logistic regression regularized with Elastic Net were tuned to $\alpha = 0.1$ and $\lambda = 0.008$. This provides a model with $Acc = 0.953$ which reaches $Acc = 0.943$ with the test data.

```
# fit model
grid <- expand.grid(C = seq(0, 1, 0.05))
```

```
svm.train <- train(model,
                   train.data,
                   method = "svmLinear",
                   trControl = trctrl,
                   preProcess = c("center", "scale"),
                   tuneGrid = grid)

# save accuracy
svm.train.acc <- svm.train$results %>%
  filter(C == svm.train$bestTune$C) %>%
  summarize(Accuracy = Accuracy)

# predict test data
svm.preds <- predict(svm.train, test.data)

# compare prediction and real data
svm.mat <- confusionMatrix(svm.preds, test.data$stroke)
```

The hyperparameter of the SVM with Elastic Net was tuned to $C = 0.05$. This provides a model with $Acc = 0.953$ which reaches $Acc = 0.943$ with the test data.

```
# fit model
tree.train <- train(model,
                   train.data,
                   method = "rf",
                   trControl = trctrl,
                   preProcess = c("center", "scale"),
                   tuneLength = 10)

# save accuracy
tree.train.acc <- tree.train$results %>%
  filter(mtry == tree.train$bestTune$mtry) %>%
  summarize(Accuracy = Accuracy)

# predict test data
tree.preds <- predict(tree.train, test.data)

# compare prediction and real data
tree.mat <- confusionMatrix(tree.preds, test.data$stroke)
```

The hyperparameter of the SVM with Elastic Net was tuned to $mtry = 2$. This provides a model with $Acc = 0.953$ which reaches $Acc = 0.943$ with the test data.

```
# list of all models
models <- list(glm.mat, reg.mat, svm.mat, tree.mat)

# function to ease extracting values from list
read.output <- function(list, info){
  if(info == "accuracy"){
    list$overall[1]
  } else {
    if(info == "sensitivity"){
      list$byClass[1]
```

```
      } else {
        if(info == "specifity"){
          list$byClass[2]
        }
      }}}

# dataframe with models' results
model.comp <- data.frame(
  model = c("GLM", "Regularized Regression", "SVM", "Tree"),
  acc = sapply(models, read.output, info="accuracy"),
  sens = sapply(models, read.output, info="sensitivity"),
  spec = sapply(models, read.output, info="specifity")
)

# create nice table
kable(
  model.comp,
  #format = "latex",
  #booktabs = TRUE,
  col.names = c("Model", "Accuracy", "Sensitivity", "Specifity"),
  align = c("l", "c", "c", "c"),
  caption = "Comparison of the Models",
  digits = 3
)
```

Table 2: Comparison of the Models

Model	Accuracy	Sensitivity	Specifity
GLM	0.943	1	0
Regularized Regression	0.943	1	0
SVM	0.943	1	0
Tree	0.943	1	0

The performance of the different models is summarized in Table 2.

Predictor Evaluation

Since the logistic regression reached highest accuracy while being the simplest model, it will be used to take a closer look at the predictors.

```
# save coefficients for text
summary(glm.train)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.0941  -0.3081  -0.1562  -0.0883   3.4865
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -4.051526   1.552319  -2.610 0.009055 **
## genderMale                    -0.005149   0.079915  -0.064 0.948630
## genderOther                   -0.160790  22.762828  -0.007 0.994364
## age                            1.719704   0.150170  11.452  < 2e-16 ***
## hypertension1                  0.081394   0.056713   1.435 0.151229
## heart_disease1                 0.077290   0.048804   1.584 0.113267
## ever_marriedYes               -0.066068   0.124156  -0.532 0.594632
## work_typeGovt_job             -0.443143   0.291175  -1.522 0.128031
## work_typeNever_worked         -0.701655  22.602517  -0.031 0.975235
## work_typePrivate              -0.557138   0.421682  -1.321 0.186426
## `work_typeSelf-employed`      -0.594527   0.321690  -1.848 0.064583 .
## Residence_typeUrban            0.134945   0.079461   1.698 0.089461 .
## avg_glucose_level              0.208859   0.062370   3.349 0.000812 ***
## bmi                            0.014112   0.100710   0.140 0.888562
## `smoking_statusnever smoked` -0.057245   0.098086  -0.584 0.559477
## smoking_statussmokes           0.070443   0.088314   0.798 0.425079
## smoking_statusUnknown          0.014158   0.108536   0.130 0.896212
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1543.2  on 4087  degrees of freedom
## Residual deviance: 1220.9  on 4071  degrees of freedom
## AIC: 1254.9
##
## Number of Fisher Scoring iterations: 14
```

```
sum.glm <- summary(glm.train)

age.est <- round(sum.glm$coefficients["age", "Estimate"], 2)
age.p <- round(sum.glm$coefficients["age", "Pr(>|z|)"], 3)

gluc.est <- round(sum.glm$coefficients["avg_glucose_level", "Estimate"], 2)
gluc.p <- round(sum.glm$coefficients["avg_glucose_level", "Pr(>|z|)"], 3)

self.est <- round(sum.glm$coefficients["`work_typeSelf-employed`",
                                       "Estimate"], 2)
self.p <- round(sum.glm$coefficients["`work_typeSelf-employed`", "Pr(>|z|)"], 3)

urb.est <- round(sum.glm$coefficients["Residence_typeUrban", "Estimate"], 2)
urb.p <- round(sum.glm$coefficients["Residence_typeUrban", "Pr(>|z|)"], 3)
```

The logistic regression with stroke as outcome yields age ($b = 1.72$, $p < 0.001$) and glucose level ($b = 0.21$, $p = 0.001$) as significant predictors.

This means, that for each year older the odds to suffer a stroke are multiplied by 5.58, and for every increase of 1 mg/dl of average blood glucose, the odds to suffer a stroke are multiplied by 1.23. To be precise, these values apply to the "average" person from the sample with average values for all variables. The possibility

that the odds change differently for people with values far from the mean cannot be eliminated with this analysis.

Although the other predictors are strictly speaking not significant, the p-value of two further predictors is close to 0.05. On the one hand, self-employment reduces the risk by 55.4% ($p = 0.065$), on the other hand urban residence increases the risk by 113.9% ($p = 0.089$).

Discussion

According to accuracy as performance criterium, there is no difference between logistic regression, regularized logistic regression, SVM and random forest. Due to the principle to keep modeling as simple as possible, as long as performance does not suffer, logistic regression is to be preferred in this case. The more complex models did not improve prediction performance.

On the first glance, the accuracy of 0.953 seems to be a good result for the goal to build a model to predict stroke occurence. Unfortunately, when considering specifity as an additional performance criterium, the prediction does not work well. Considering the data, the reason for this probably lies in the ratio from subjects with to subjects without stroke which is 1:19.52. This ratio leads to an unbalanced dataset, and accuracy can be a misleading performance criterium: Because of the high portion of healthy subjects, high accuracy is easy to acquire by always predicting "no stroke." There are solutions for unbalanced datasets like Over- or Under-Sampling which unfortunately go beyond the scope of this project work.

Although the created models will thus not work well to predict future stroke cases, the predictors can still be interpreted. The logistic regression presented age as a factor which multiplies the odds to suffer a stroke by 5.58 for each passing year. This is not unexpected, as the prevalence of many diseases increases with age and stroke is among the most common causes for people 65 years of age and older (Sahyoun et al. 2001). As second predictor, the average glucose level emerged by multiplying the odds of a stroke by 1.23. The glucose level is tightly-knit to the diagnosis of type 2 diabetes which is known to increase stroke risk as well (Sander, Sander, and Poppert 2008).

In addition to the significant predictors, the other variables are also interesting to answer which factors are rather not relevant to assess the stroke risk. Here, demographic and medical data seem to be more important than lifestyle data, though the results for residency type and self-employment are interesting. It is possible that some of the other, not-significant variables correlate with age and thus could not contribute more than age already did.

Limitations

The limitation of this work lies primarly in the unbalanced dataset, which impeded the model construction with the known methods.

Additionally, the validity of the data is not certain, though no variables or values stood out in an all too negative way. Also referring to metadata, it would have been useful to know how the data where collected, especially in which city, region, country or culture. With this lack of information, generalization of the results is limited.

Future Research

Future research should try to work around the aforementioned limitations, i.e. use methods to transform the data into a more usable, balanced dataset. Another possibility would be to use different data that is more balanced to begin with.

Although a heterogenous dataset that represents the general population well can be important for classification/prediction research, it can also be of use to work with a specific subset to identify risk factors of this

subset. It would even be useful to identify which risk factors apply to everyone and which are more relevant for specific subgroups.

Conclusion

Unfortunately, the original goal of this project work to build a well-working stroke prediction model could not be achieved due to the unbalanced data. Nevertheless, age and glucose level could be identified as important predictors.

References

Darehed, David, Mathias Blom, Eva-Lotta Glader, Johan Niklasson, Bo Norrving, and Marie Eriksson. 2020. "In-Hospital Delays in Stroke Thrombolysis: Every Minute Counts." *Stroke* 51 (8): 2536–39.

fedesoriano. 2021. "Stroke Prediction Dataset." https://www.kaggle.com/fedesoriano/stroke-prediction-dataset.

Kassambara, Alboukadel. 2020. *Ggpubr: 'Ggplot2' Based Publication Ready Plots.* https://CRAN.R-project.org/package=ggpubr.

Kuhn, Max. 2020. *Caret: Classification and Regression Training.* https://CRAN.R-project.org/package=caret.

Liaw, Andy, and Matthew Wiener. 2002. "Classification and Regression by randomForest." *R News* 2 (3): 18–22. https://CRAN.R-project.org/doc/Rnews/.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

RStudio Team. 2020. *RStudio: Integrated Development Environment for r.* Boston, MA: RStudio, PBC. http://www.rstudio.com/.

Sahyoun, Nadine R, Harold Lentzner, Donna Hoyert, and Kristen N Robinson. 2001. "Trends in Causes of Death Among the Elderly." *Aging Trends* 1 (1): 1–10.

Sander, Dirk, Kerstin Sander, and Holger Poppert. 2008. "Stroke in Type 2 Diabetes." *The British Journal of Diabetes & Vascular Disease* 8 (5): 222–29.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, and Dana Seidel. 2020. *Scales: Scale Functions for Visualization.* https://CRAN.R-project.org/package=scales.