What is the most accurate model in predicting the stroke risk of a person given 10 attributes.

Orfeas Gkourlias

2022-10-18

Introduction

This document aims to explore, analyse and explain the data set being used in answering the following research question: "What is the most accurate model in predicting the stroke risk of a person given 10 attributes?". As the res earch question implies, the data set consists of 10 attributes. This project has the goal of comparing those attributes, so that the most likely predictors for a stroke may be deduced. Some attributes affect each other, while others may not. Analysis of these correlations can help in finding the rankings of the attr ibutes. The final answer to the research question will be found using Weka, a data platform which utilizes machine learning.

To get a feel for what the scope and attributes of the data set consists of, it will be loaded and the first 10 results will be displayed.

```
main <- read.csv("../data/stroke-data.csv")
head(main)
```

```
##       id gender age hypertension heart_disease ever_married     work_type
## 1  9046   Male  67            0             1          Yes       Private
## 2 51676 Female  61            0             0          Yes Self-employed
## 3 31112   Male  80            0             1          Yes       Private
## 4 60182 Female  49            0             0          Yes       Private
## 5  1665 Female  79            1             0          Yes Self-employed
## 6 56669   Male  81            0             0          Yes       Private
##   Residence_type avg_glucose_level  bmi  smoking_status stroke
## 1          Urban            228.69 36.6 formerly smoked      1
## 2          Rural            202.21  N/A   never smoked      1
## 3          Rural            105.92 32.5   never smoked      1
## 4          Urban            171.23 34.4         smokes      1
## 5          Rural            174.12   24   never smoked      1
## 6          Urban            186.21   29 formerly smoked      1
```

```
nrow(main)
```

```
## [1] 5110
```

There are 12 attributes, 10 of which will be used in the analysis: Gender, age hypertension, heart_disease, ever_married, work_type, residence_type, avg_gluco se_level, bmi and smoking_status. The last column indicates whether the person has already experienced a prior stroke. This can be used to the train the machine learning model which will be utilized to answer the research question.

There are 5110 entries in this data set. This is also why the row numbers will not be replaced with the id's, because there is no order in the id numbers. They exceed the number 5110.

The attributes and their units can be seen in the code book on the next page.

Codebook

```
knitr::kable(codebook)
```

Column	Unit	Description
ID	Number	Unique patient identifier
Gender	Text	"Male", "Female" or "Other"
Age	Number	Age of patient
Hypertension	Boolean	Whether patient has hypertension
Heart_disease	Boolean	Whether patient has a heart disease
Ever_married	Boolean	Whether patient has ever been married
Work_type	Text	Occupation status of patient
Residence_type	Text	Patient living enviroment
Avg_glucose_level	Number	Average glucose level in blood
BMI	Number	Body mass index of patient
Smoking_status	Boolean	Whether patient smokes or not
Stroke	Boolean	Whether patient has ever experienced a stroke

Initial Data and Attributes

In this section, the attributes will be examined individually. What these attributes could mean for the research question will be discussed. Correlations will be observed in a later section. Any preprocessing or cleanup required will also be performed in this section.

ID

This column is neither noteworthy for analysis or data structure. This column will therefore be dropped, because the dataframe used already has row numbers and this makes the ID redundant.

```
main <- main[2:12]
```

Age

The age of the patient. At first sight, it might look redundant for this data to be stored as a float, since most of the data consists of a rounded age number. Some of the entries contain very young patients. The younger a patient is, the more important the specificity of the age is, since the age difference is still significant at that point. It is for that reason that any patient under the age of 2 will contain a float number, with two decimal numbers. A couple of those instances will be shown in vector format below:

```
head(c(main[main$age < 2, 2]))
```

```
## [1] 1.32 0.64 0.88 1.80 0.32 1.08
```

The likelihood of a person experiencing a stroke increases with age. This will therefore be an important attribute in the analysis.

Gender

Gender, containing three separate values: Male, Female and other. There could be a correlation in the gender and stroke risk of a person. An example of gender indirectly affecting the stroke risk of a patient would be the age difference in genders. Women tend to live longer lives than men, therefore the stroke risk may increase for women, because they generally become older. This could be one of many hypothesis.

```
main$gender <- factor(main$gender)
```

Hypertension

This indicates with a 0 or 1 whether the patient is affected by hypertension. The first attribute which is relevant to the heart status of a patient. These types of attributes will always be important, because any heart condition tends to come with an increased risk of experiencing a stroke. Since this is indicated with a binary number, the patient either has hypertension, which is indicated with a 1, or not, which is indicated with a 0. Correlations will probably be found between hypertension and the other attributes. In this case, it will be easier to work with "Yes" or "No" values. Not only for the sake of consistency, but because Weka otherwise recognizes the 0's and 1's as numeric values, not boolean values This will therefore be changed with the following code:

```
main$hypertension <- factor(main$hypertension, labels = c("No", "Yes"))
```

The datatypes were also changed to that of a factor. This has been done so that later functions will properly work on the data.

Heart Disease

Similar to the previous attribute, this is also an important element when trying to predict stroke risk. The previous observation also applies to this attribute.

```
main$heart_disease <- factor(main$heart_disease, labels = c("No", "Yes"))
```

Ever married

This displays whether the patient has ever been married in their lifetime. This will most likely not bet detrimental in predicting the stroke risks of patients. But this is part of the data set, so it will therefore be compared with the other attributes, to see where it ranks with it's prediction.

```
main$ever_married <- factor(main$ever_married, labels = c("No", "Yes"))
```

Work Type

A similar attribute to the prior one. Will most likely not be a good predictor for stroke risk. But it may rank higher than the marriage attribute. Some sectors could theoretically expose a person to environments where strokes are more likely.

```
main$work_type <- factor(main$work_type)
```

Residence Type

Considering that some types of residency might be healthier than others, this attribute may be slightly important in determining the stroke risk of a person.

```
main$Residence_type <- factor(main$Residence_type)
```

Average Glucose Level

This may be more important than the prior three attributes, especially when these levels are unusually low or high. The literature concerning glucose levels and their connection to strokes is still being debated. Some papers conclude that it is not detrimental when observed in non-diabetic people.

BMI

The body mass index is an indicator for how a person's weight/height ratio. Age and gender also being taken into consideration for the calculation. Both extreme ends of this attribute could be important to the stroke risk of a person. Higher BMI levels are also associated with developing heart disease. Glucose levels may also be affected. Several other attributes are most likely going to have a correlation to this attribute.

Smoking Status

Whether a patient is smoking will most likely affect some of the other attribute s in this data set. Whether these significant correlations will need to be tested. The smoking status is unknown for some of the patients.

```
main$smoking_status <- factor(main$smoking_status)
```
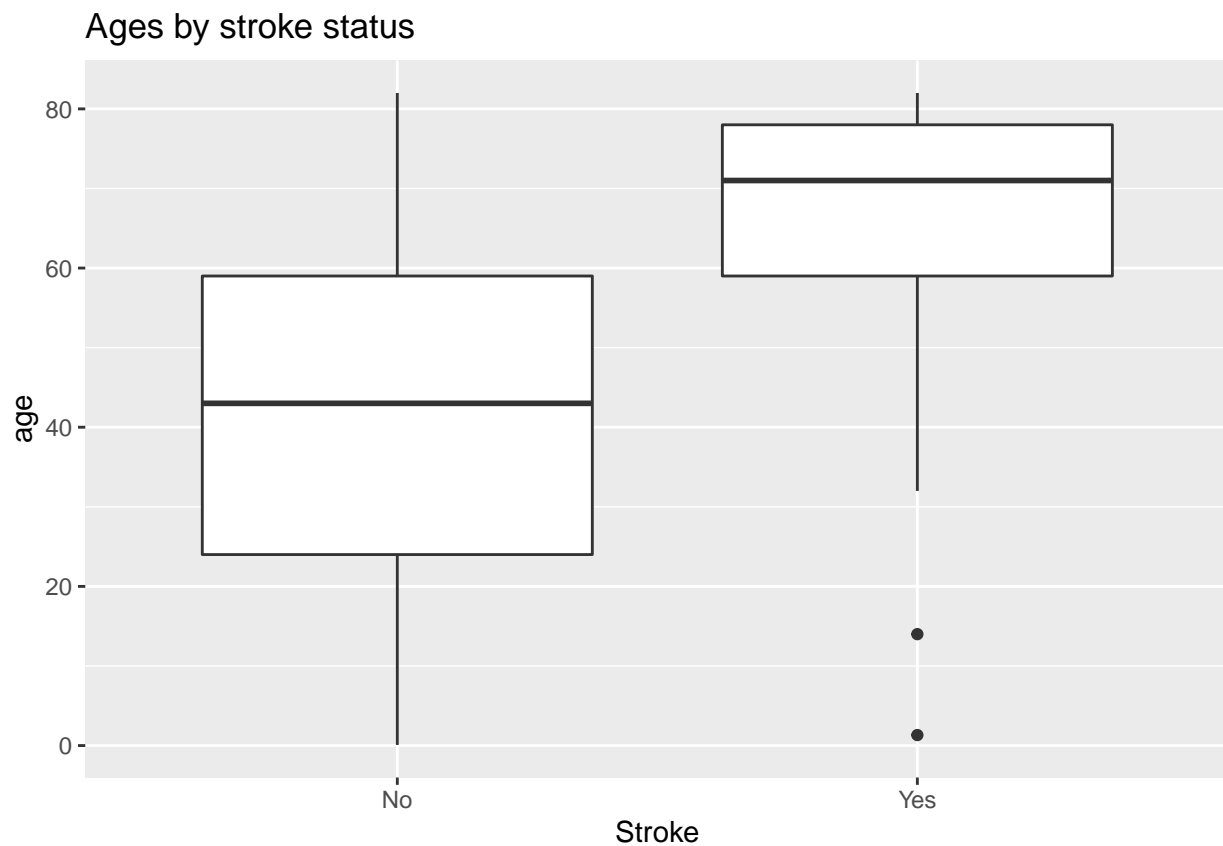
Stroke

The column indicating whether the patient has ever had a stroke. This is the classifier which will be predicted for in the final model, with the highest possible accuracy.

```
main$stroke <- factor(main$stroke, labels = c("No", "Yes"))
```

Correlations

Following the examination of the initial values and attributes, they may now be compared so that trends and correlations can be observed. Starting with the first attribute, which will most likely be important in determining stroke risk, age. By plotting the summaries of patients who had a stroke, and those who did not, maybe a link can be seen between the two.

```
no_stroke <- main[main$stroke == "No",]
stroke <- main[main$stroke == "Yes",]
ggplot(main, aes(x=factor(stroke), y=age))+geom_boxplot()+ggtitle("Ages by stroke status")+
  xlab("Stroke")
```

Ages by stroke status

This plot shows that the group of people who have had strokes are, on average, older than the group who has not experienced a stroke. This may also be observed with a summary

```
summary(no_stroke$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.08   24.00   43.00   41.97   59.00   82.00
```

```
summary(stroke$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.32   59.00   71.00   67.73   78.00   82.00
```

It is worth noting that the no stroke group does contain plenty of patients of higher age.

Now, a t-test may be performed to determine whether age is a significant difference between the two groups. A one sample t-test is most appropriate here, considering that every patient's risk is independent from another

```
t.test(no_stroke$age, stroke$age)[3]
```

```
## $p.value
## [1] 2.115685e-95
```

Without considering the other attributes, it seems that with that p-value, the two age means are significantly different.

These tests may be performed for all of the other attributes too. Showing them all individually would be redundant, since the correlations between having a stroke and the other attributes relating to the heart already have literature confirming them.

The attributes, which may seem less influential at first would be worth checking manually. The marriage status being the first one to consider. This column is not a boolean number, even though it is comparable to the other 1 or 0 columns. For consistency, yes will become 1 and no will become 0.

```
no_stroke <- main[main$stroke == "No",]
stroke <- main[main$stroke == "Yes",]

no_s_table <- log(table(no_stroke$ever_married))
s_table <- log(table(stroke$ever_married))

vec <- c(no_s_table[1], no_s_table[2], s_table[1], s_table[2])

barplot(vec, space = c(0,0,1,0), col = c("brown4", "cadetblue"),
        names.arg = c("                              No Stroke","",
                      "                              Stroke",""), ylab = "Log(Occurance)")
legend("topright", c("Not Married", "Married"), fill = c("brown4", "cadetblue"))
```
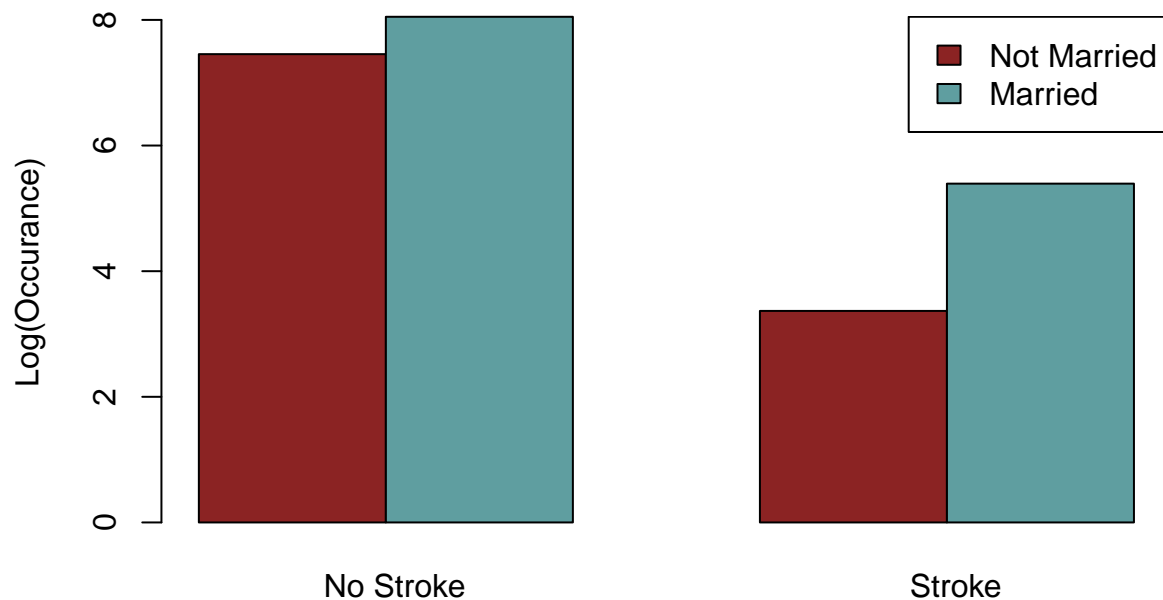

The data has been normalized, so that it may be properly displayed. The group of people with no stroke quite lower than the other group. This plot shows that for both groups, the amount of people who have ever been married is bigger than the not married group. It makes sense for there to be proportionally more people who have married than less when looking at the stroke bars. This is because people who have had a stroke are on average older than the other group. It is also safe to assume that the older a person is, the more likely that they have ever been married in their life.

Now, it is possible to make a similar plot, but with a more relevant attribute. Heart disease would be a good choice.

```
no_s_table <- log(table(no_stroke$heart_disease))
s_table <- log(table(stroke$heart_disease))

vec <- c(no_s_table[1], no_s_table[2], s_table[1], s_table[2])

barplot(vec, space = c(0,0,1,0), col = c("brown4", "cadetblue"),
        names.arg = c("                        No Stroke","",
                      "                        Stroke",""), ylab = "Log(Occurance)")
legend("topright", c("No Heart disease", "Heart Disease"),
       fill = c("brown4", "cadetblue"))
```
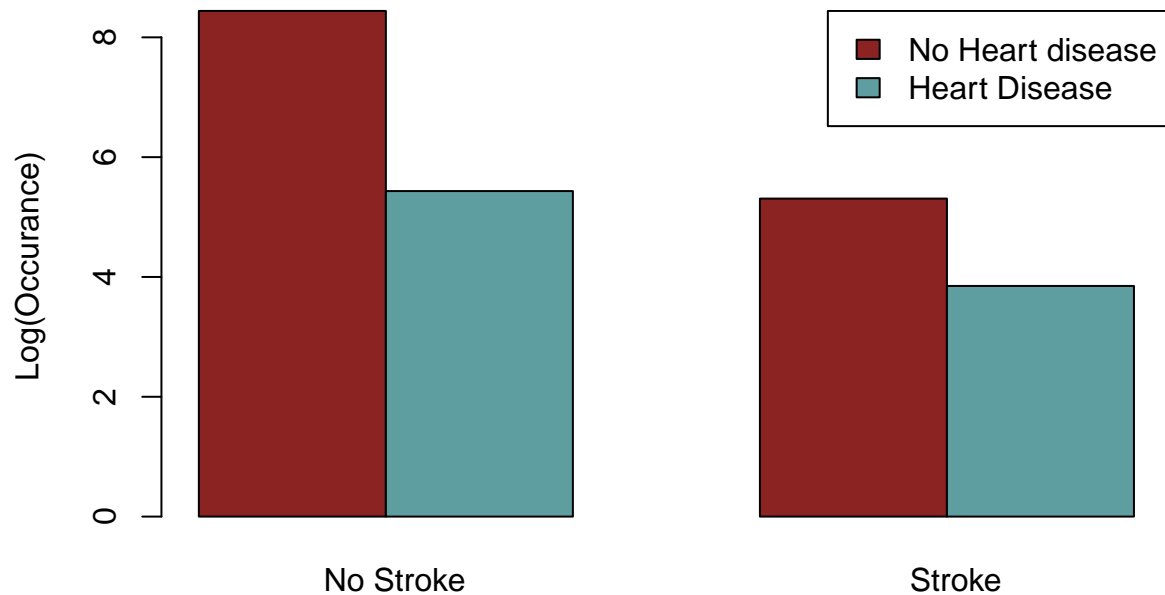
The assumption would be that a heart disease has a correlation with having a stroke. The bar plot above does not show anything out of the ordinary. It might be difficult to find any correlation by just looking at these bar plots. Calculating the differing odds between two groups could help.

```
ratio1 <- as.vector(table(no_stroke$heart_disease)[2])/nrow(no_stroke)
ratio2 <- as.vector(table(stroke$heart_disease)[2])/nrow(stroke)
ratio1
```

```
## [1] 0.04710965
```

```
ratio2
```

```
## [1] 0.188755
```

In this case, it does appear that the ratio is different. A calculation can be made to determine how much more likely patients with heart conditions may get a stroke.

```
ratio2/ratio1
```

```
## [1] 4.006717
```

Judging by that simple calculation, a patient with a heart condition has 4 times the likelihood of getting a stroke than someone without a heart condition.

A bar plot showing the correlation between BMI and other attributes could also show correlations. Before doing that however, it seems that the BMI column consists of characters, not numbers. This must first be changed.
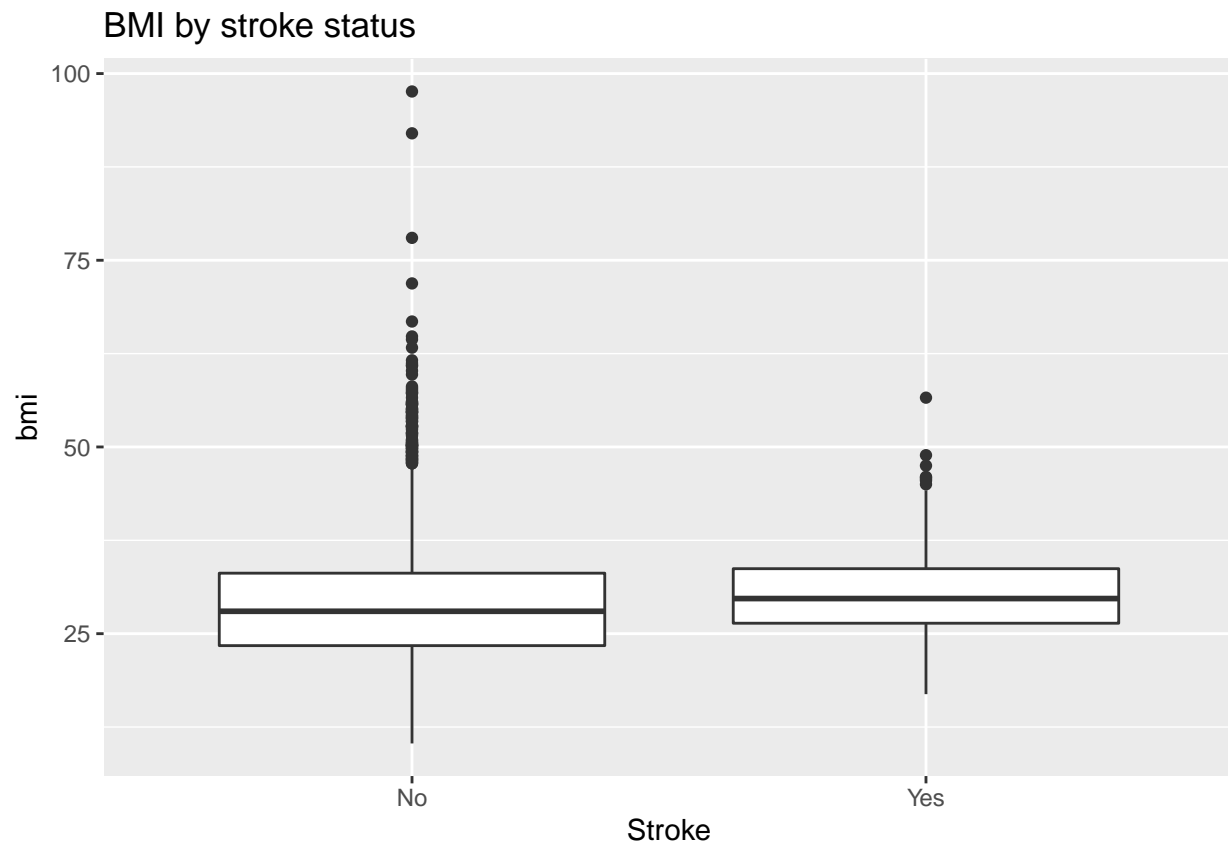
```
typeof(main$bmi)
```

```
## [1] "character"
```

```
main$bmi <- as.numeric(main$bmi)
```

```
## Warning: NAs introduced by coercion
```

```
ggplot(main, aes(x=factor(stroke), y=bmi))+geom_boxplot()+ggtitle("BMI by stroke status")+
  xlab("Stroke")
```

```
## Warning: Removed 201 rows containing non-finite values (stat_boxplot).
```


There are around 200 NA entries, which were not taken into consideration when creating the plots.

Now that the main data set has also been cleaned up, it will be written to a .csv file, so that it may be used in the journal.

```
path_out = "../data/"
write.csv(main, paste(path_out, "main.csv", sep = ""), row.names = FALSE)
```

Weka Model Exploration.

The data set can now be put through weka algorithm testing. The goal here is to train an algorithm which will have the most accuracy when it comes to stroke prediction. RWeka is a package which allows for Weka procedures to be performed in R.

The main file is currently a .csv file. Weka works better with .arff files, so it will therefore be copied to an .arff format for future use.

```
RWeka::write.arff(main, file = paste(path_out, "main.arff", sep = ""))
```

Data imbalance and ZeroR

Something else which is of high importance is the data imbalance. To properly train the model, there need to be more entries where the stroke class indicates "Yes". Otherwise, the ZeroR algorithm will net the highest accuracy, which makes sense with the current imbalance. 95% of the data entries have not yet had a stroke. ZeroR will therefore offer the highest accuracy of 95% with this balance.

It is for that reason that an oversampling technique was chosen to help combat this problem. Synthetic Minority Oversampling Technique (SMOTE) will be used to this end. SMOTE can be performed using the DMwR package.

It is important the training data is not affected by SMOTE, but the test data may be modified.

```
set.seed(101)
sample = sample.split(main$stroke, SplitRatio = .75)
train = subset(main, sample == TRUE)
test  = subset(main, sample == FALSE)
smote_df <- SMOTE(stroke ~., test, perc.over = 1800, perc.under = 200)
table(main$stroke)
```

```
##
##   No  Yes
## 4861  249
```

```
table(smote_df$stroke)
```

```
##
##   No  Yes
## 2232 1178
```

The tables show the difference pre and post SMOTE application. This will also be written to an .arff file for Weka.

```
RWeka::write.arff(train, file = paste(path_out, "train.arff", sep = ""))
RWeka::write.arff(test, file = paste(path_out, "test.arff", sep = ""))
```

Performing ZeroR on this new arff file gives an accuracy of 65%. This is still an imbalance, but not as severe as before. Models may now be trained using different classifiers.

NaiveBayes

Performing NaiveBayes on the SMOTE dataset provides an accuracy of 81.05%. This is already more accurate than simply performing a ZeroR.